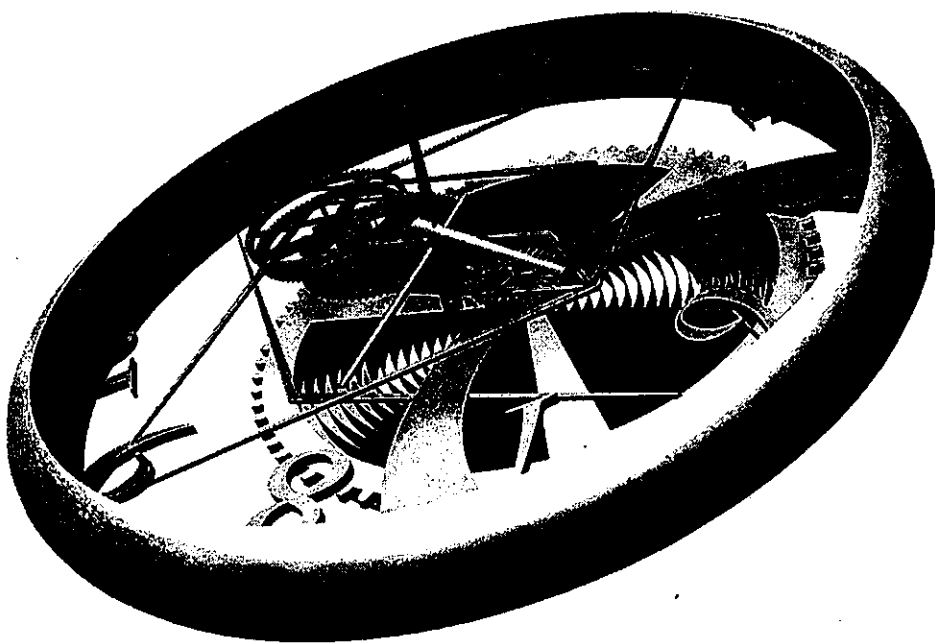
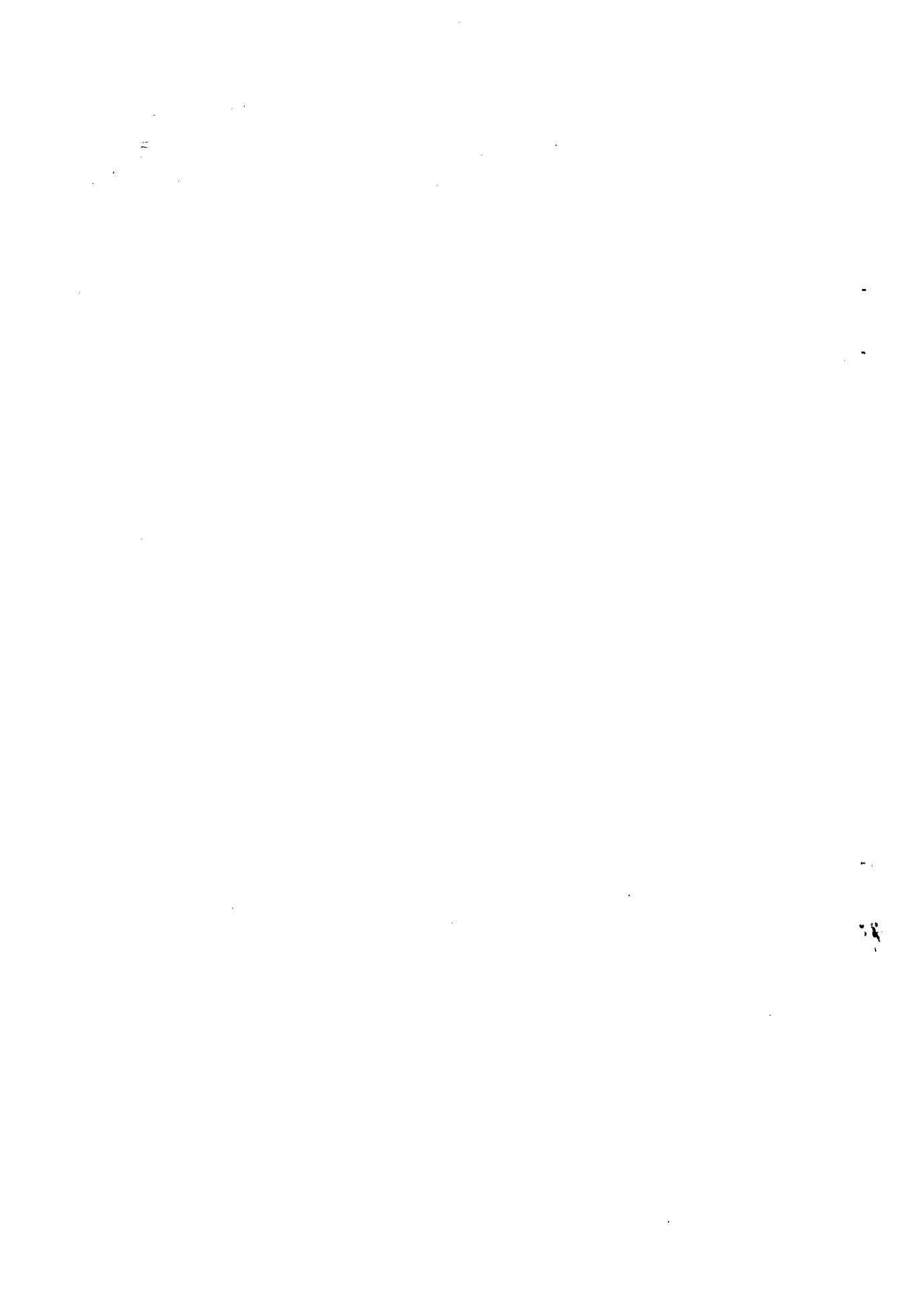


2013 A-FE

PHYSICS

CLUB





はじめに

昨年、回路図集では「A-P
と呼びますが、最近ではよ
りあります。スル一ルを微細化したよ
CPUが発表されていす。
高性能が追う32bitマイコンに
我が部でも液晶の制御が一般
よるSDや回路集は麻布学
化してきていす。この「回路図集」は、部員らが
園物理部無線班の研究、製作さ
年をかけたす。ただし、未
のものも含まれるおり、必ず
のしも本動作する保証はあり
まなく動かし始め承下
人。あらかじめ承下

会計：栗本 郁也

Contents

はじめに P. 01

- コンピューター系 -

F-PAD P. 05 H2 梅津 直弥
BLUE-BALL P. 15 H1 水野 太郎

- ロボット系 -

Mogix3 P. 21 H2 新田 京太郎
蜘蛛の目 P. 27 H2 栗本 郁也
H1 金子 真太郎
M3 朽木 良輔
Alto P. 35 H1 河村 洋一郎
T6 P. 40 H1 岸田 聖生
M3 佐藤 惇
南風に靡く歯車 P.46 M3 花園 佳月

- ゲーム系 -

物無負トローゼ P. 50 H2 石原 皓
M3 黒田 健太
SpaceStoneShooter P. 56 H1 山本 涼一
M3 四柳 雄太

花火師、ロロンを飼う。	P. 62	M2	石倉 匠
母船を守れ	P. 64	M2	市川 詠亮
土竜ノ叩き	P. 69	M2	佐伯 大地
Space Aegis	P. 72	M2	藤川 拓巳
SNAKE	P. 74	M2	宮嶋 優大
塵袋との科学大衝突	P. 78	M2	横尾 幸丸
さよなら夏の日	P. 81	M2	山本 涼太郎
背水の陣	P. 84	M2	渡辺 悠太
イライラ棒	P. 88	M2	佐伯 大地
		M2	渡辺 悠太
DEATH STAR2	P. 91	M2	石倉 匠
		M2	横尾 幸丸

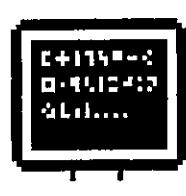
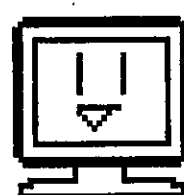
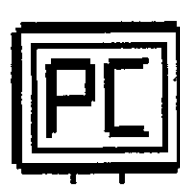
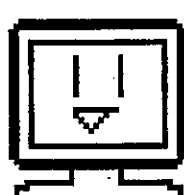
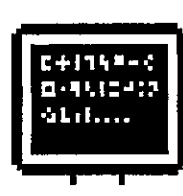
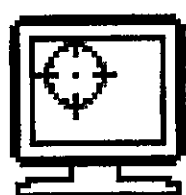
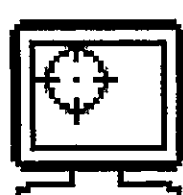
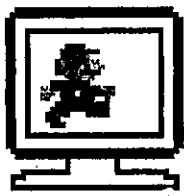
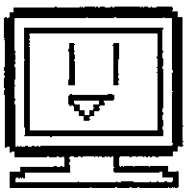
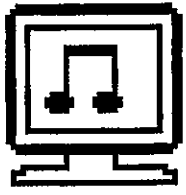
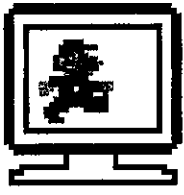
- 販売物 -

BTURIS	P. 94	M3	佐藤 惇
チカチカ	P. 98	M3	黒田 健太

Web

azabubtm.web.fc2.com/





PC MODE



H2 梅津 直弥

協力・感謝:物理部およびネット世界の皆様
そして各種プログラムの開発者様等

概要:

2+1 画面で構成されるマルチメディアプレイヤーです。

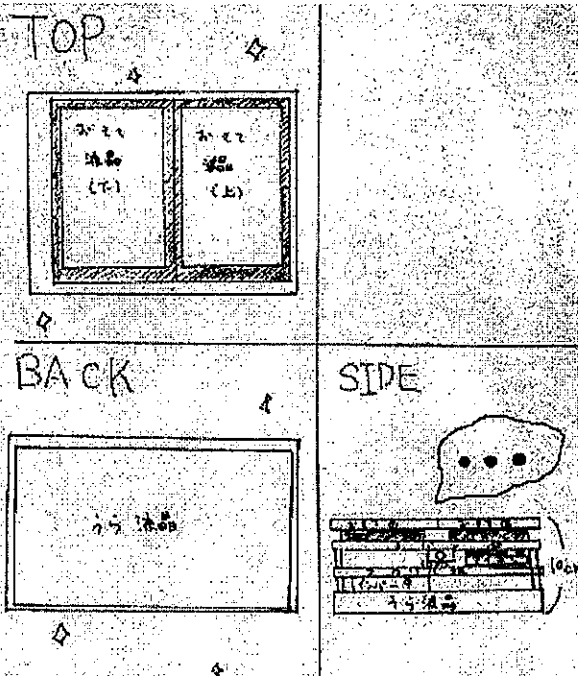
PAD 端末...というには後述のとおり無理のある体型。

2画面で構成される面(以降、おもて面とします)の内1画面が2012年4月1日現在タッチパネルに対応しており、直感的動作が可能です。

裏面の7インチ液晶ディスプレイではおもて面比べて高速で複雑な動作が可能です。よりレスポンス性が要求されるアプリケーションや、大画面での表示が適するアプリケーションはこの液晶で実行することで最適な環境での動作が可能です。

外部端子としては micro-SD カードコネクタと 3.5mm イヤホン端子が備えられています。ただし、現段階では micro-SD カードをシステムブートメモリとして使用している関係上、取り外した状態での起動は不可能な状態です。将来的にはより大容量なシリアルフラッシュメモリを用意することで SD メモリはこの制約から解放されます。ちなみに、USB 接続端子も搭載。2012年4月1日現在では未だ制御に至っていないため、未対応となっております。対応すれば、USB メモリからのデータの読み取りや、キーボード接続による入力が可能となります。

概観・外観:



- ・合計三枚もの液晶を贅沢に使用
 - ・表に二枚
 - ・裏に一枚
 - ・表の二枚のうち片方がタッチ対応
- ・挟んで何と10センチ。
- ・重量:およそ5・600グラム
- ・総工費:10000円くらい(一部は去年の流用)
- ・総製作期間:8カ月間
 - ・有効製作時間:1000時間ほど
- ・搭載 CPU:SH72620
- ・動作周波数:144MHz
- ・内蔵 RAM:1MHz
- ・キャッシュ搭載
- ・電源供給:DC12V
- ・使用液晶:
 - ・WBX280V009 *2
 - ・L0070T5DR06 *1
- ・使用インバータ:INV2L-03VS
- ・使用 MP3 デコーダ:VS1011e



仕様:

- CPU部 //////////////////////////////////////
CPU部は Renesas 社 SHT262 マイコンを搭載。14.4MHz という超高速(ホビーコースでは)で動作し、内蔵 RAM も 1MB と大容量(こちらもホビーコースでは)と、物理部史上(きっと)最強の“頭脳”です。Interface 誌の付録、キャッシュまでついて至れり尽くせり。雑誌の企画で取り上げられている分得られる情報は多い。これ自体は多分専用に作られたのかな?
- MP3 に関して //////////////////////////////////////
MP3 は VLSI ソリューションの VS1011E を使用。SHT262 は DAC(デジタルデータをアナログデータにコンバートするもの)を非搭載なので、何か代替りのものとし使用。正直な話、MP3 デコードは CPU だけで行える(更にデコードにおいて DAC の代わりに出来る PWM もあるけどね)のですが、資料の多いこちらを選びました。(より機能の多い上位品種があるのにとが言ってはいけません。)音質はよくも悪くも普通。
- 表の液晶 //////////////////////////////////////
二枚ある液晶はショッピングサイト & その直営店で販売されているパラレル接続タイプの液晶を使用。WBX280V009 と言いますが、ちょっと古い型で今では取り扱われていません。ただしほぼ互換品が購入可能。これらの液晶の特徴は、内蔵されたビデオメモリにデータを書きこむだけで表示ができるため、マイコンのスペックが低くても高画質な画像を表示できることと、人気があり比較的情報を得やすいことです。幾つかのサイズがあり、おおよそ互換性があります。
理論上こちらの液晶でもそこそこ高速な描画は可能なのですが、マイコンの LCD コントローラを生かしてあげたいので割愛します。なお、本来ならばタッチパネルは上下とも搭載されておりますが、タッチパネルは非常に割れやすい部品です。そのため二枚とも無事ということがないのです…何枚割ったことが。タッチパネルは非常に割れやすいので注意。
- うらの液晶 //////////////////////////////////////
裏面の液晶も同上的お店で販売されています。このタイプの液晶を制御するためのコントローラがマイコンに内蔵されているため、より高速で複雑な描画が可能です。物理部液晶史上(多分)最高画質の WQVGA、±10V という気持ち悪い電圧が必要なのですが、三端子レギュレータと電圧コンパターで対処。バックライトは LED ではなく冷陰極管なので、aitendo にて販売されているインバータ使用。
- 使用した SD カードについて //////////////////////////////////////
サンディスク製の 2GB のマイクロ SD を使用。
今回 SD カードに使用しているポートは、マイコンボード上のシリアル FlashROM のデータ出力とぶつかってしまうので、シリアル FlashROM からのプログラムブート時はリレーで機械的に絶縁しています。アホっぽいですがおそらく IC だと遅延のせいか正常な通信が出来なかったのです。
なお、当然のごとくファイルシステムは FAT、FatFs にはたいへん大きなお世話になりました。(ファイルシステムを扱うためのモジュールです。めっちゃくちゃ便利。)大変感謝。
今回、SD 制御に関して、デフォルトの設定ではライセンスが必要(というか仕様さえ教えてくれない)なため、簡単な制御方法として有名な SPI モードを使用しています。なぜかオーバークロック気味の 36MHz で動いている。
- 使用した電源について //////////////////////////////////////
スイッチング電源を使用。そこそこコンパクト。電源電圧は 12V です。これは、最も高い電圧を必要とする液晶のバックライト制御回路に合わせた値です。余裕を持って 2A 給電できる物を使用しています。

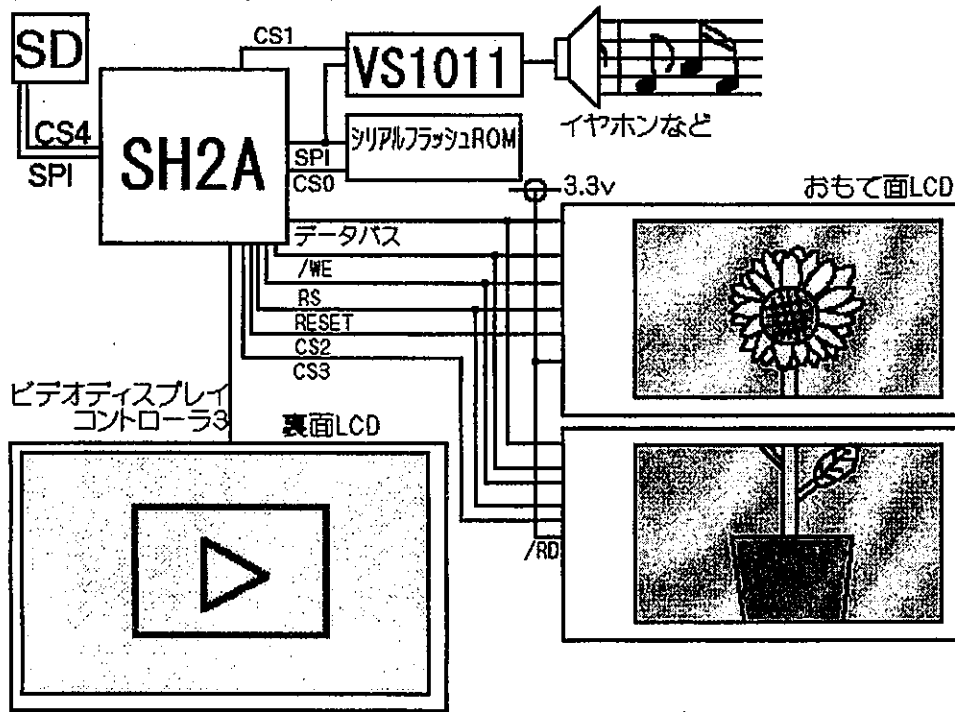
開発環境:

制作にあたり使用した環境群です。

- ・パソコン :HP ProBook 4515s Windows7 home premium
型落ちした普通のパソコン
- ・統合開発環境 :High-performance Embedded Workshop 評価版 Version 4.08.00.011
評価版なので無料だが、コンパイラに制限があって辛い
- ・コンパイラ :GNUSH v11.02 Windows Tool Chain (ELF) MP1
正規コンパイラの辛い制限を解決してくれるフリーのコンパイラ
- ・デバッグ :HJ-LINK/USB
安価な JTAG デバッグ。CPLD などコンフィグできる。
- ・デバッグソフト :HJ-LINK/USB Debugger for SHT262
デバッグを使うためのソフト。変数の読み書きなど自由自在だが、変数書き込みでのブレークはできない模様。



おおまかなブロック図：



※あくまでも「大まかな」ものです。正確な接続方法については各パーツメーカーより公開されているデータシートの方をご覧ください。

※離れてしまっておりますが SPI 信号線は全て共通です。

※各ポートの割り当てはアドレスピンを GPIO として使ってる等割りと滅茶苦茶なので割愛させていただきます。

※ピン数が多いマイコンなので、回路図を1ページで収めると潰れて見えなくなってしまうので掲載致しませせん。

諸注：

CS…チップ・セレクトの略。この信号を送ることで「お前働け」「お前寝てろ」と命令できる。

SPI…シリアル・ペリフェラル・インタフェースの略。

少ない信号線でデータを高速にやり取りするための規格。

WE…ライト・イネーブルの略。この信号でデータの書き込みのタイミングを指定。

RE…リード・イネーブルの略。この信号でデータの読み出しのタイミングを指定する。今回は未使用。

RESET…そのまま。動きもそのままです。初期化時に使用します。

供給電圧：

表液晶:3.3V 単一

三端子レギュレータで生成

裏液晶:3.3V、5V、10V、-10V

三端子レギュレータ/電圧コンバータで生成

インバータ:12V 単一

DC アダプタから直接供給

マイコン:5V 単一

ただしボード上で I/O 用・コア用など最適な電圧に変換

デコーダ:3.3V 単一

三端子レギュレータで生成

Sd カード:3.3V 単一

三端子レギュレータで生成



使用したピンの配置表：

ピン番号	主な機能	F-pad での接続先	ピン番号	主な機能	F-pad での接続先
1		タッチパネル/XL	107	LCD_DATA12	うら液晶/R2
4		タッチパネル/XR	109	LCD_DATA11	うら液晶/R1
5		タッチパネル/YU	111	LCD_DATA10	うら液晶/G5
7		タッチパネル/YD	112	LCD_DATA9	うら液晶/G4
11	A1	コントローラ/↑	113	LCD_DATA8	うら液晶/G3
12	A2	コントローラ/↓	114	LCD_DATA7	うら液晶/G2
13	A3	コントローラ/→	115	LCD_DATA6	うら液晶/G1
14	A4	コントローラ/←	116	LCD_DATA5	うら液晶/G0
15	A5	コントローラ/A	117	LCD_DATA4	うら液晶/B5・B0
16	A6	コントローラ/B	119	LCD_DATA3	うら液晶/B4
17	A7	コントローラ/X	121	LCD_DATA2	うら液晶/B3
19	A8	コントローラ/Y	123	LCD_DATA1	うら液晶/B2
21	A9	SD カード/D0 絶縁用リレー	124	LCD_DATA0	うら液晶/B1
23	A10	VS1011/RESET	127	MISO0	SPI/IN
24	A11	VS1011/XDCS	129	MOSIO	SPI/OUT
25	A12	VS1011/DREQ	131	SSL00	SPI/CS
26	A13	VS1011/XCS	133	RSPCK	SPI/CLK
27	A14	表液晶/CS(上)	154	D15	表液晶/DB17
28	A15	表液晶/CS(下)	155	D14	表液晶/DB16
30	A16	表液晶/RS	156	D13	表液晶/DB15
34	A17	表液晶/WR	157	D12	表液晶/DB14
35	A18	表液晶/RD	158	D11	表液晶/DB13
36	A19	表液晶/RESET	159	D10	表液晶/DB12
82	AN0	タッチパネル/XL	162	D9	表液晶/DB11
83	AN1	タッチパネル/YU	163	D8	表液晶/DB10
98		SD カード/CS	164	D7	表液晶/DB7
99	LCD_CLK	うら液晶/CLK	165	D6	表液晶/DB6
100	LCD_DE	うら液晶/ENAB	166	D5	表液晶/DB5
101	LCD_HSYNC	うら液晶/HSYNC	167	D4	表液晶/DB4
102	LCD_VSYNC	うら液晶/VSYNC	168	D3	表液晶/DB3
103	LCD_DATA15	うら液晶/R5・R0	170	D2	表液晶/DB2
104	LCD_DATA14	うら液晶/R4	172	D1	表液晶/DB1
105	LCD_DATA13	うら液晶/R3	174	D0	表液晶/DB0

※正確なピン配置や機能表はメーカーの最新データシートをご覧ください。

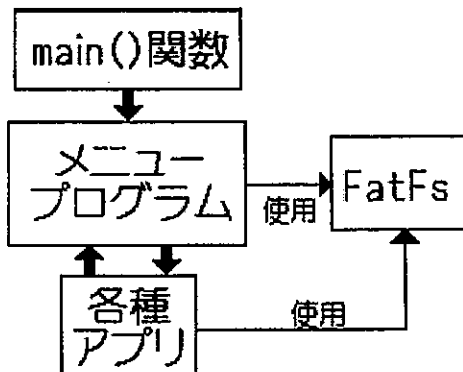
※表記通りに製作された時の動作を保証するものではありません。

※ボードの関係上 CPU のすべてのピンが使用可能ではありません。



プログラム構造 :

起動後まず最初に main()関数が呼び出され、アプリケーションを起動するための最低限の初期設定を終え、メニュー画面へと移動します。メニュープログラムと各種アプリケーションはSDカード内に保存されたファイルにアクセスするためにFatFsを使用します。呼び出されたアプリケーションがタスクを終えメニュー画面に戻り、ユーザーがまたアプリケーションを選択して、のループです。



搭載アプリケーション :

- ピクチャー**////////////////////////////////////
micro-SD カード内の Picture ディレクトリ内の画像ファイルを取得してファイル名を下画面に一覧表示。その後、ユーザーがタッチで選択して、上画面に表示されます。
なお、上画面を使用しないアプリケーションでは画面が切り替わらないため、ちょっとした壁紙になります。
- ミュージック**////////////////////////////////////
micro-SD カード内の Music ディレクトリ内の MP3 または WAVE 形式のファイルを取得してファイル名を下画面に一覧表示。その後、ユーザーがタッチで選択します。
選択後は曲操作画面に切り替わり、曲への操作が可能。曲一覧へは画面端のボタンで戻れます。
- 時計**////////////////////////////////////
時計を表示します。閏年も対応、腕時計機能も完備です。ただ、電源を切ると時計はリセットされてしまいます。文化祭の展示用なので、アジャスト機能に華を持たせるためにこの機能は割愛します。
- ペイント**////////////////////////////////////
簡易ペイントです。直線や円などある程度の描画が可能です。パレットは 10 色です。画像の保存機能は、展示の際に勝手に保存されそうなので取ってつけてませんでした。
- 色あわせゲーム**////////////////////////////////////
実は去年の制作物の流用。ただし、劣悪だった操作面は大幅に改善。
ほぼランダムに選ばれた色がお題として出され、赤、緑、青の三色のバランスをあわせてお題と同じ色を作るゲーム。難易度の関係で答え合わせはフルオートで実行。二色合っていればあと一色をいじるだけで正解になるものの、中々うまく行かないのが面白いところ。
- 電卓**////////////////////////////////////
「とってつけたゴミ電卓。プログラムを思い切り手を抜いたので精度が甘い。6桁まで対応なあたりからも製作者のやる気が伺える。」
欠点がないと可愛くないということで欠点をつけたらそれは「欠陥」になった。致命的な演算能力を誇る、物理部屈指のゴミ電卓。しかし、普通の電卓にはない面白さがそこにはある。どうぞお楽しみください。
- ピアノ**////////////////////////////////////
ある意味最もタッチパネルらしく、最も「アプリ」らしいアプリケーション。ファミコンライクなサウンドを奏でます。
音色はデューティ比 60% の方形波。2 オクターブ対応で、上下に分割しています。
- 肉焼きゲーム**////////////////////////////////////
どことなくデジャブを感じるゲーム。ぎりぎりまで火を通して上手に焼きましょう。ちよびり残酷なのがなんとも言えない。シンプルな操作性が一層何とも見えぬ何かを醸し出します。
- シューティングゲーム**////////////////////////////////////
レイヤ機能などを活用したおもて面液晶での縦スクロールシューティングゲームです。所謂弾幕系が否かと言えますが、処理体型的関係上高度すぎる描画はできませんので、「ちょっと敵弾の多いスローライフなシューティングゲーム」といったところ。コントローラ(と言っても付け合せ程度に作った)使用。後述。



動画再生////////////////////////////////////
 micro-SD カード内の movie ディレクトリ内の動画ファイルを取得してファイル名を下画面に一覧表示。その後、ユーザーがタッチで選択します。選択後は真画面で動画を再生。動画一覧へは付属コントローラのいずれかのボタンで戻れます。BMP の連番画像を順に読みだして力押しの現状。ライセンスとか面倒臭いので望みは低い。

その後の拡張性：

今回は行っておりませんが、プログラムを変えれば SD カードにアプリケーションを保存し、自動で検索して一覧表示するといったことも可能です。展示で受けを狙うにはコストパフォーマンスが悪すぎるので後回しに.. 動画表示もなんとか組み込みたいところ。

また、USB 機器との接続もプログラム次第で対応可能です。でもやっぱりマウスを使った操作よりタッチペンでの操作のほうが受けがよさそうなので後回しに..

「高度なプログラム」が「すごい」制作物に必ずしも結びつかないのが嘆かわしいところ。

シューティングゲーム：

せっかくなので出来る限り本格的なシューティングゲームを作ろうということで作ったゲーム。せっかくなので他の物無負よろしくそれっぽいストーリーも、どこかで有りそうな話だ。

ストーリー：

-----20XX 年-----

近年突如として現れた、“アポカリプス”と名付けられた惑星。

しかし、研究が進むにつれ、次のことがわかってきた。

それは、その惑星が原始的な“意思”を持つことと、

重力を操り、他の惑星を吸収して成長する性質を持つことだった。

さらに、地球周辺に重力に由来すると思われる異変が確認された。

その後重力の異変は近年急速に進み、地球の公転さえ影響を受ける程になった。

最悪の事態を回避すべく開発された無人宇宙戦闘機“Hawke”に人類の希望が託された今、

壮絶な惑星アポカリプス破壊任務が始まったのだった…

ゲーム画面（おおよその再現）：



実際の画面ではシンプルながら可愛いグラフィックになっているのですが、ここではどうも安っぽく映ってしまっているのが非常に残念です。当然、実物はカラーですよ。



システム：

遊び方：

いきなりボス戦。敵弾をかわくぐりつつ、自分の攻撃を当てて、地球の平和を守れ！敵の体力を完全に0にすることができたらゲームクリアです。

コントローラー：

コントローラーは十字キーにA・B・X・Yボタンと直感的な配置。

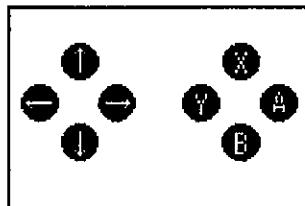
左半分のボタンで上下左右に移動。

Aボタンでボムを使用。

Bボタンで攻撃。

Yボタンで装備変更。

Xボタンでリペア。



外形図→

攻撃と装備：

Bボタンを押すことで弾を発射して攻撃します。

Yボタンを押すたびに装備を切り替えます

・ノーマル

一直線に進む黄色い弾を発射します。一見なんの取り柄もない装備ですが、ボムエネルギーの溜りが格段に早いです。

・3WAY

3列同時に弾を発射します。弾の進む方向は一直線です。

その分攻撃力は上がりますが、ボムエネルギーの溜まる速さは遅くなってしまいます。

・レーザー

一直線に進むレーザー砲で攻撃します。

非常に高い攻撃力を有しますが、装備中は「ゲームバランスがさすがに悪いので」動力機関にまで負荷が及ぶ(という設定の)ため、移動速度が下がってしまいます。

また、ボムエネルギーの溜まる速さも非常に遅いです。

・ホーミング

標的を自動追尾するホーミング弾を発射します。

横方向に推進力を調整することは可能ですが、縦方向の調整はできないため、敵との位置取りによっては当たらないこともあります。(そんなに接近すること自体稀なのでまず当たらない)

ボムについて：

ボムのストックが溜まっている場合、Aボタンを使用してボムを使用することができます。

時期の周辺を周回する二対のレーザーを発射します。

噴らいボムはありませんが、代わりにYボタンを押すことで、ボムを二個消費して残機を1増やす「リペア」ができます。なお、ボムによる攻撃中は完全に無敵で、敵弾を攻撃でかき消すこともできますが、制限時間があるので注意してください。

ボム以外の名称も考えたものはいまいなかったため結局「ボム」に。

ミスとゲームオーバー：

敵弾に被弾するとミスとなり、残機が1減ります。残機が0の時にミスをするとうゲームオーバーです。

敵弾パターン

敵のライフゲージが青色

自機を囲むように6つの弾が現れます。落ち着いて横に移動することで対処できます。

敵のライフゲージが水色

自機の方向に向かい円形に陣を組んだ弾が現れます。弾速は早いです。落ち着いて移動すれば対処できます。

敵のライフゲージが緑色

あらゆる場所からあらゆる方向にランダムに弾が出現します。直接自機の周りに出現することは無いので、きちんと見極めましょう。動かないほうが安全か、動いたほうが安全かを瞬時に判断する必要があります。

敵のライフゲージが黄色

真下に向かって進む弾を、発射地点を左右に揺らしながら3列に撃ってきます。規則性の塊のため正直余程のことがない限りミスり得な



簡単なパターン。ボムエネルギーをしっかりと稼いでやりましょう。
敵のライフゲージが赤

青色の時と黄色の時の攻撃パターンを同時におこなってくる上、敵は自爆により自機を
道連れにしようと自機を吸い寄せてきます。的確な操作が要求されます。貯めていたボ
ムの使い所。勝利は目前です。気を緩めないように。

BGMについて：

micro-SD カード内の STG ディレクトリに保存された“BGM.mp3”を読み出して使用。あまり激しい描
画が続くと処理落ちにより音が飛んでしまったりします。ゆくゆくは CPU への負荷の少ない方法
で実装する予定。

グラフィックについて：

シンプルな丸のほうが可愛かったので丸にしましたが、別途 micro-SD カードに画像を保存すれ
ば自機や敵、更には弾の画像も変えることができます。もちろん背景もこの手法で変更が可能で
す。

犠牲者の墓標：

このコラムは製作過程において犠牲となった部品たちを弔い次は二度と殺さないように懺悔するコラム
です。南無。

WBX280V009	*3	タッチパネルを割られるなり、液晶ごと割られるなり。慎重に扱きましょう。 タッチパネルは一部の場所が特に割れやすいので特に注意。
SH2A マイコン	*1	液晶との接続を改善中に謎の死を遂げる。接続の調整は電源を切ってからやりましょう。
micro-SD カードソケット	*1	場所が気に入らなくて剥がされる際にはんだこての熱で変形。 はんだ吸い取り線等を利用して丁寧に剥がしましょう。
各種ピンヘッダ & ソケット	*無数	基盤をリニューアルする際に剥がそうとして失敗。 場所が確定する前は最悪の場合でもはがしやすいように最小限のハンダ付けにしましょう。
各種コンデンサ・抵抗	*もっと無数	場所や容量/抵抗値を間違えたがためにゴミ箱へ。 勿体無いのできちんと考えて部品を選びましょう。
三端子レギュレータ	*1	爆熱したので放熱機構を付けるため、壊して剥がす。 剥がし方にも作法はあります。
USB ケーブル	*1	必要と思って使うために切った矢先代替となる部品が出現。最早言うことはない。
USB コネクタ	*1	同上の理由でお役御免に。
イヤホンコネクタ	*1	DIP 化キットがあったのでそっちに取り替え。 上位品の確認は常に行いましょう。
配線材	*数十本?	ここでもそこでもない、短すぎ長すぎ、と、一番振り回された部品。 一度回路図を起こしてからハンダ付けをはじめるといいでしょう。

断っておきますが、無駄にしようとしていたわけではありませんよ。皆さんも部品を大事に扱うように。



参考文献及びサイト：

FatFs

http://elm-chan.org/fsw/ff/00index_j.html

FatFsに関連する数々のサイト様

Interface (CQ出版)

2010年6~12月号 2011年1~4月号

付録 SH2A 基盤に関連する数々のサイト様

VS1011E に関係する数々のサイト様

両種類の液晶に関する数々のサイト様

以上そして全ての情報提供者様(僕が使用すると想定されてなかった方も)に感謝を申し上げます。

部品調達先：

液晶…aitendo.cc

インバータ…aitendo.cc

SH2A マイコン…若松通商

10V 三端子レギュレータ…千石電商

木材…島忠

コの字アルミ材…東急ハンズ

マイクロ SD カード…あきばお〜

その他…秋月電子通商

感想：

まず後輩に：

物理部無線班での制作において重要な者の一つが年度を超えての下積みです。僕も去年からのノウハウを生かし、まあ他の物無負を見ても上々な出来の部類には入れたのではないのでしょうか。

唯一の心残りは共同制作を出来なかったこと。何をやるにも打ち合わせが必要なのはめっちゃくちゃ面倒だけど、喜びを分かち合える相手が欲しかった。(←部員がいるので十分)悔いのない物理部生活を送ってください。よく言う話、全力さえ尽くせば、たとえ「無駄だったかなあ？」と思うようなことがあっても「あんなに熟中できたなあ」とジーンと来るはず。

僕としてはデバッグの導入もかなりおすすめだよ！変数の値を見る(本当に超絶便利、書き換えることさえ可能)とか、特定の場所で止めるとか、バグの原因を突き止める強力な兵器だ。高いけどね！RXマイコンの純正デバッグが12K位だそうだ。安い方安い方。(コンパイラがどうかはまだ知らんが)

一般的に：

制作物名の謎の文字である、F が Final の頭文字である通り、これが僕の現役最後の制作物です。あんまり凝ったネーミングにするのは性に合わないのでベターなラインに落ち着いたわけです。

単は関係ないこととしておきます。

コアな技術にこだわり過ぎると、残念なことに例えそれがどんなに高度な技術でも展示の際に「一般受け」に欠けてしまうためあまり「高度なプログラム」を書くことにはこだわりませんでした。どちらかと言えば力押しのプログラムです。時間があつたらもっと色々なことをやりたかったです。僕としてはこのマイコンは浪漫にあふれたスベックなので、余す所無く活用したかったのですが、どうにも時間と技量が足りないわけで、後輩にはぜひ RTOS とかやってみてほしい。

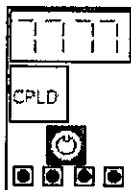
とは言えきっと、自分の中では今までで一番すごい制作物ができたと思います。そして、今までで一番周りのみんなの力を借りたと思います。泣けるエンディングにしたいのは山々なのですが、これと言った締めくくる言葉が見つからないのが非常に残念ですが、ご精読、ありがとうございました。



その他の制作物：

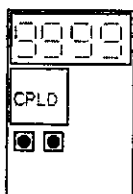
ひっそりとCPLDで製作したゲームについてさりと説明します。
全部回路図での記述で製作しました。Xilinx の ise webpack 9.2i を使用。
書き込みは impact と cblsrv と HJ-LINK USB を使用。各開発者様に感謝です。

・スロットゲーム



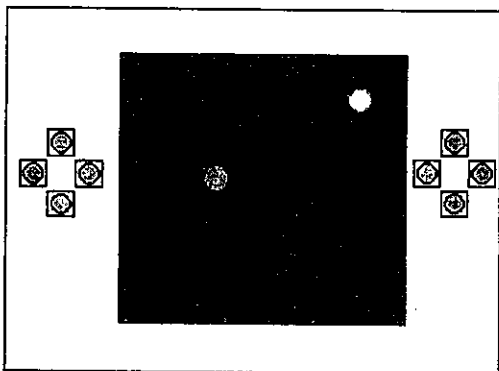
使用CPLD:xilinx xc9572xl
中身はカウンタの塊。
4けたの7セグを使用したスロットゲームです。
下部の四つのボタンを押すとストップ。
見事すべてを"7"で止めることができればクリア！
真ん中のつまみで難易度調整。自押しができるのはレベル9くらいまで。
裏のボタンでリセット。

・チキンレースゲーム



使用CPLD:xilinx xc9572xl
こちらも中身はカウンタの塊。
4けたの7セグを使用したチキンレースゲームです。
二つあるボタンのうち右側のボタンでゲームスタート！
左側のボタンでストップ！ぎりぎりまで粘って"9999"に近づきましょう！
オーバーするとゲームオーバー！ぶるぶる震えます。

・鬼ごっこゲーム



1・2P の区別はあまり意味がありません

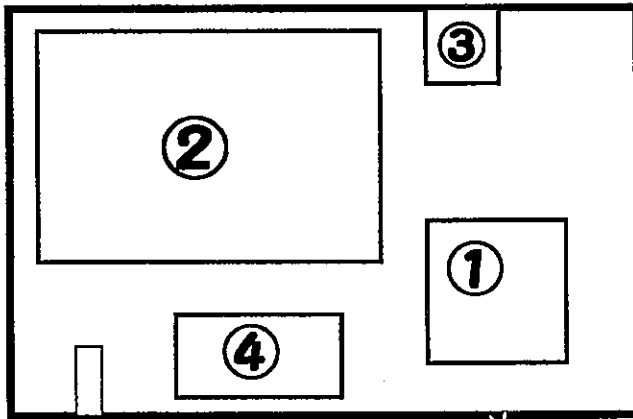
使用CPLD:xilinx xc95288xl
中身はカウンタやマルチプレクサなど少し複雑。
8*8の二色ドットマトリクスLEDを使用した鬼ごっこゲームです。
ルールは簡単、鬼が逃げる側と重なればOK！
移動は各プレイヤーの四つのボタン。上、下、左、右。
ただし、鬼は一定時間ごとに1Pと2Pで交代するので注意。
敵に追い詰められても、がんばって時間いっぱいしのげば反撃のチャンス！
逆に、追い詰めたとこで攻防逆転してしまえば大ピンチ！

BLUE-BALL

製作:H1 水野太郎

協力:物無の皆さん

○外観○○○○○○○○○○○○○○○○○○



① マイコン

② 液晶

③ MicroSD

④ VS1011e

詳しくは制御で

○ゲーム○○○○○○○○○○○○○○○○○○

①おはなし

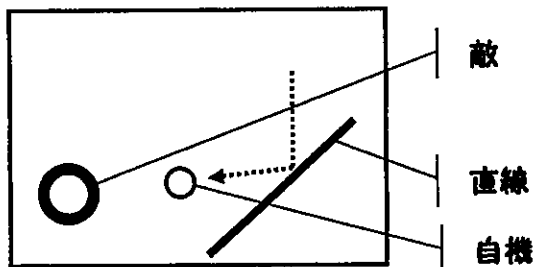
ここはのんびりとしたむら、「ミニビレッジ」。あおいうみとすなはまが広がるなんごく。そこにすむ人々はうみのむこうがわの「ライトシティ」ですてられ、なみにのってながれてくるがらくたを大切にひろいあつめ、それをつかって生活しています。

ある日、いつものようにすなはまでがらくたをさがしていたむらの少年「りょういち」は、あおいボールと赤いぼうを見つけました。あおいボールを追っていくといつのまにか迷ってしまいました。泣きそうになる「りょういち」の前にスーツの男があらわれました。「おやおや、迷子ですか。かわいそうに... わたしがおうちへつれていってあげましょう」。人をうたがったことなどない「りょういち」はその男についていきました。気がつくともふねにのせられ、いつ

のまにか大都会”ライトシティ”についていました。そこでやっただまされたど
 気づいた”りょういち”は、すきをついて男から逃げました。”りょういち”は赤
 いぼうとあおいボールしかもっていません。これだけをつかって、恐ろしい”
 ライトシティ”からおうちのある”ミニビレッジ”に帰らなければいけません。”
 りょういち”のぼうけんのはじまりだ！

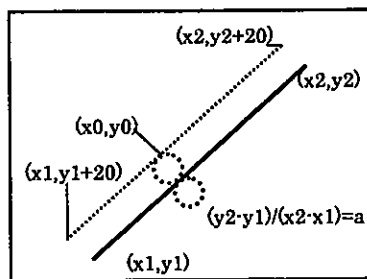
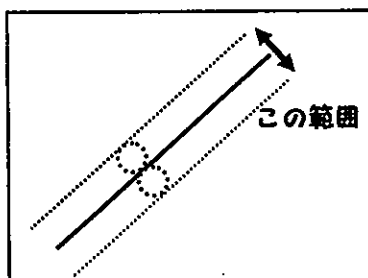
② ゲーム説明

このゲームは、自機で敵を倒しながら進みゴールを目指すゲームです。タッ
 チペンを画面に付けた所と離した所に直線が引かれます。その直線に画面上で
 動く自機が当たると反射するので、それを使って自機を進ませていきます。画
 面上の敵を全て倒すと次のステージに進み、最後のボスを倒すとゲームクリア
 です。ホームボタンを押すことでゲームを強制終了することができます



③ 反射方法

まず直線と自機が当たったかの判定は自機が左図のような時に起こるよう
 にして計算は右図より $(y_0 - y_1 > (x_0 - x_1) * a) \&\& (y_0 - y_1 < (x_0 - x_1) * a + 20)$ となります。



反射の計算は元の直線の傾き A_1 、自機移動の傾き A_2 より、反射後の傾き A_3
 を求めるために、 $\tan(\text{atan}A_2 * 2 - \text{atan}A_1) = A_3$ としています。ただ特別なパタ
 ーンである、直線が水平、垂直の場合はそれぞれ y 値、 x 値を反転しています。

○アプリ○○○○○○○○○○○○○○○○○○

どうせなので軽い他のアプリも作ってみました。まあ梅津さんには敵わないのでそこはご了承ください。

①ギャラリー

サムネイルが表示されるのでそれをタッチすると画像が全体に表示されます。二番目のボタンを押すとサムネイル画面に戻ります。

②ミュージックプレイヤー

タッチで曲を選べます。一時停止、再生ができます。

○制御○○○○○○○○○○○○○○○○○○

①マイコン

今回使ったのは PIC32MX440f512h というマイコンで処理速度が他の PIC より命令速度が速い 32MX シリーズを使っています (dspic とかの二倍ぐらい)。速度に不満はなかったのですが、ピン数が 64 では若干足りなかったので、100 ピンのものを使えばよかったなと少し後悔しています。

②液晶

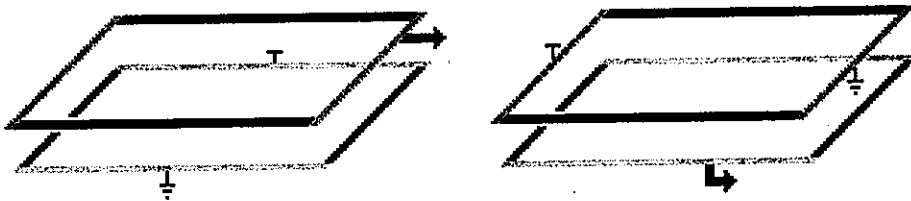
Aitendo というお店で取り扱われている、CP2401T (2.4 インチ) を使用しました。制御といっても後閑さんが制御した YHY024006A と動作が全く同じなのでそれを参考にしました。表示したいデータを内部に一時保存ができるので、データをずっと送り続ける必要がないのが売りです。しかしデータの送信が遅いため、表示に無駄な時間がかかってしまいます。

タッチパネルについて

静電容量方式 (iphone とか)、電磁誘導方式 (ペンタブと同じ原理)、超音波方式 (大きい液晶に使われるらしい、一番この方式がカッコいいと思う)、... など数多のタッチパネル制御方式がある中で、この液晶では抵抗膜方式でタッチ検出をすることができます。抵抗膜式は制御方法が比較的に簡単であるため、少し

説明をさせていただきます。。

抵抗膜方式では名前の通り、液晶の表面上に透明な抵抗膜が2枚貼られています。そしてそれぞれ1枚目の左右、2枚目の上下に電極が付いています。まず2枚目の電極にHとLをかけます。そして上からタッチするとタッチ部分で1枚目と2枚目がショートします。するとタッチ位置が上下方向のどこにいるかによって1枚目の電極に流れる電圧が変わります。次は1枚目の電極にHとLをかけ、タッチをします。すると今度は左右方向のどこにいるかによって2枚目の電極にかかる電圧が変わります。これによって上下方向でどの位置、左右方向でどの位置がタッチされているかがA/D変換することによってわかります。



抵抗膜式タッチパネルは上記の手順を素早く行うことで、いつタッチしてもその値がすぐに出るようになるため、本制作物ではタイマ割り込みを使って50msec毎にタッチ判定を行なっています。

③MicroSD

これに画像、音楽データを入れ、必要なときに読み出しています。SPI通信によってデータの受信を行っています。

④VS1011e

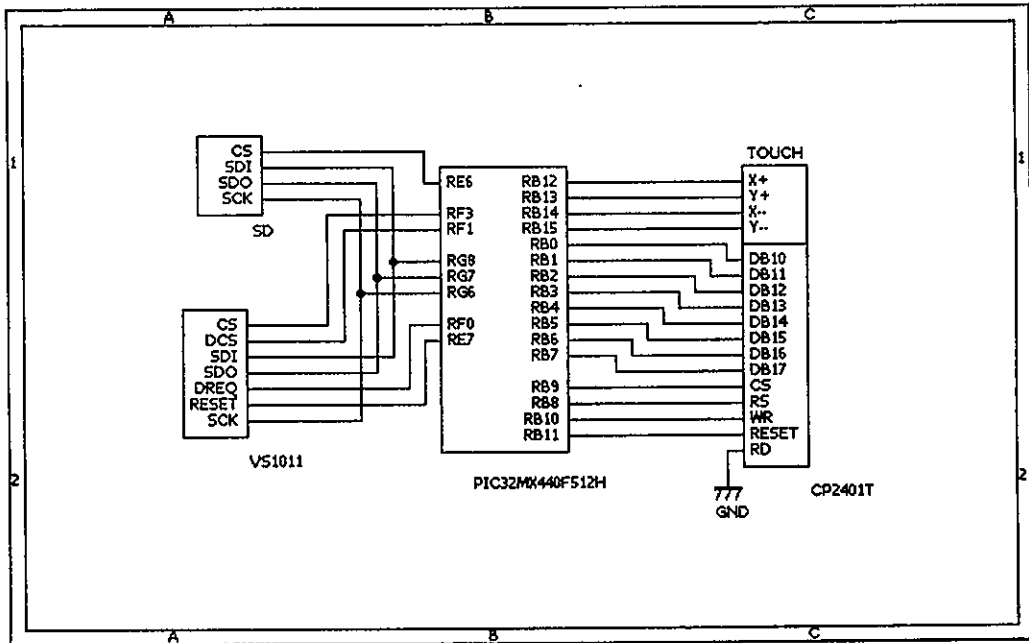
mp3データをデコードして音楽を流すことができるICです(実は一部のwavデータも流すことが出来るらしいです)。内部は制御部とデータ受信部にわかれ、それぞれXCS、XDSピンでどちらとSPI通信を行うかを決めます。制御部では初期化、CLKの速さ、音量音質の設定が行えます。

SPI通信について

細かく説明する訳ではないのですが、下の回路図を見て「あれ、なんでSPI通信のピンがSDにもVS1011にもつながってる。こん

なんじゃ信号が混ざって壊れちゃうよ」と思った人用に。SD との通信をするときは SD の CS を L にして、VS1011 との通信をするときは VS1011 の CS、DCS のピンを L にすることで通信が片方ずつでしか起こらないようにしています。

○回路○○○○○○○○○○○○○○○○

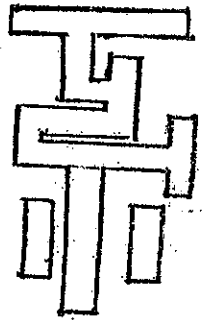
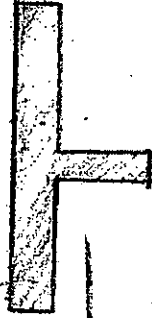
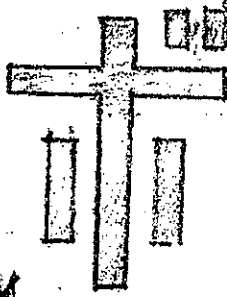
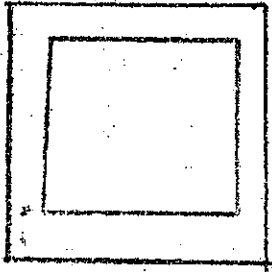


※電源や、PCでの書き込みピン、発振器などを省略しています

○感想○○○○○○○○○○○○○○○○

去年は途中で売り物に阻まれたので、好きなもので製作ができたのは今年が初めてだった気がします。なので去年よりはモチベーション高く望めた気がします。しかし、最後まで先輩に頼りきりだったと思います。特に梅津さんには何度も熱心に教えていただきました、ありがとうございます。制作自体は遅れてはいいませんがやはり安心できる位置では無いです。なので動いていない反射プログラムを早く動かして、余裕を作っていきたいと思っています。





今は昔、大和の国に、
昔ならぬ願いを持てる者あり。
闇の垣間から人間の汚さを知り、
世界の不条理に吞まれる。
数多の人間を知り、
世界の平和に身を賭して、
闇をも屈したという。
其の者を人は、————と呼ぶ。

然う、
少年は世界の不条理から、
世界の不平等から、
平和を勝ち取るため、
平等を勝ち取るため、
世界の覇者となることを誓った。
巨大ロボを作り、
人々に夢を手え、
世界を駆け巡ることを夢見た。

Mogixはその過程の一つである————

Mogix 3

製作者：H2 新田 京太郎

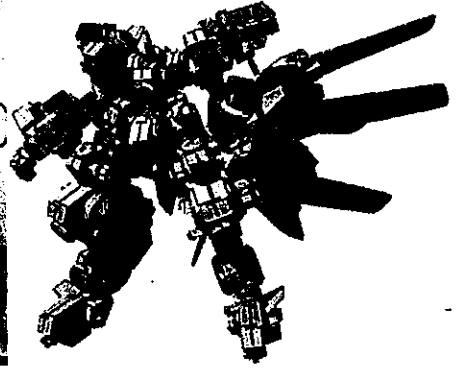
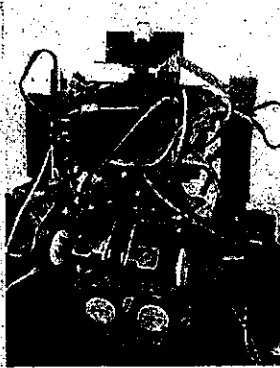
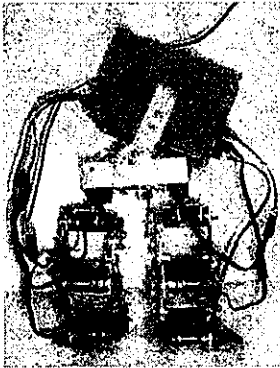
協力者：物大のアリエッティ

Secret 1. Outline

世界を支配するロボットとして、まず軽やかな足取りを目指し、無駄なモーターを省いた単純かつ小さい設計の二足歩行ロボットです。去年の構造より改善がされています。

Secret 2. Information

	Mogix (去年制作)	Mogix 3 (今年制作)	Mogix X (近未来製作予定)
最高速度	0.6cm/s	0.6cm/s	299 792 458 m/s
関節自由度	2×4	2×5+ 2×2+1	∞
電源	5V	5V	核融合
重量	600g	200g	10000000t
高さ	30cm	20cm	634m
操縦	自動	無線	内部乗り込み型
外観			



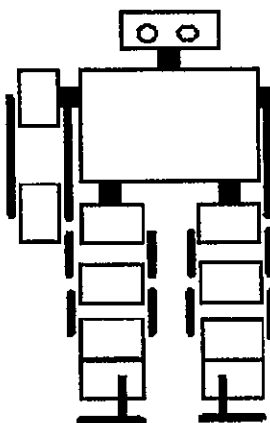
動画



by ミスター・ブシード

ちなみに、Mogix 2 は挫折。構造は、Mogix 3 のサーボモーターの配置で Mogix(初代)のモーターを使おうとして破綻。重すぎた。

Secret 3. Walking

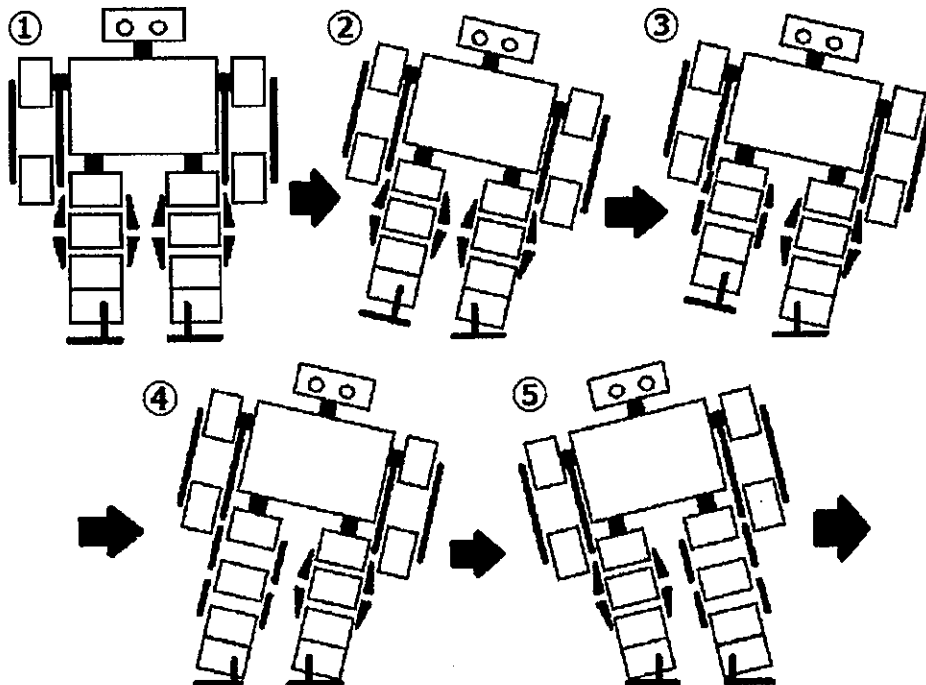


今回は特別に歩行モーションを記そう。

右の絵は立っている時の Mogix 3。しかし、この体勢をすることは、足を完全に伸ばしていると、足を出せない。なので、以下の①のような前かがみが基本体制となる。

なお、白い小さな四角(計 12 個)はサーボモータで、中心の大きな四角は中に回路が入っている。黒い太線は金属板でサーボを止めているものと考えてもらいたい。

それでは、歩行モーションに入る。



①は下から 3 番目のサーボ 2 つを前に出してしゃがんでいる状態である。この体勢が最も安定している。

②は①から右の足首のサーボモータを傾けた体勢である。右足を浮かすために体を傾けている。つまり、この時は左足だけで立っている。

③は左足を前に出したところである。

④は左足を地面につけた状態。左足は右足より前に出ている。

⑤は、足を両足につけたまま、体の傾きを変えたものである。左足は地面に接しているが、重心はほぼ右足にかかっている。その次は、②の左右対称型。つまり、左足を上げて右足だけで立つモーションに入る。以下、同じように左右対称で動作を続ける。

Secret 4. Catching

Mogixの手にあたる物。

使うもの：サーボモータ、1Ω抵抗

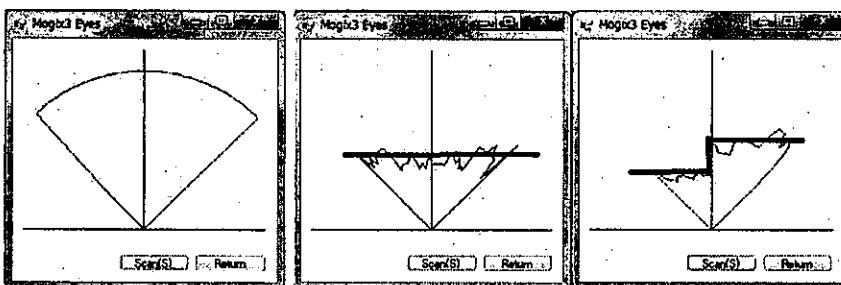
ただモーターを使った手を作ってものを握ろうとすると握りつぶしてしまうので今回サーボモータの特徴を利用して握りつぶさないように改良しました。サーボモータは指定角度まで回れない（つまり物を掴んでいて、回りたいが物理的に回れない状態）時、消費電流が上がるといった特性があります。つまり、サーボモータの消費電流を監視していれば、手が物を掴んでいるというのがPICで認識できるというわけです。回路は簡単で、サーボモータの0Vの前に1Ω抵抗をはさみ、その抵抗の前の電圧を測ります（PICのAD変換）。 $V=IR$ の方式を使い、 V （求めた電圧） $=I$ （消費電流） R （今回は1Ω）で消費電流を割り出します。

Secret 5. Watching

Mogixの目にあたる物。

使うもの：赤外線測距モジュール『GP2Y0A21YK』（秋月）

目の前の障害物までの距離を測定します。将来機には、レーザー光線で破壊する機能を搭載します。今回のMogix 3は頭部分にこのモジュールを付け、首が回るように設計したので、前方約90度の範囲の障害物を検知することが出来ます。下はモジュールの信号をパソコンで画像にしてみたものです。左から、目先に何も無い状態。真ん中は壁を認識している状態（黒線はあとで書きました。モジュールが安物なので大体です）。右は壁の前に更に障害物がある状態。実際はこんなふうな画像としては出ませんが、この情報を下にPICが処理をして壁に近づいたらそれ以上歩かないようにします。ちっちゃい障害物だと腕を使いどかします。



Secret 6. Standing

Mogixの平衡感覚を司るもの。

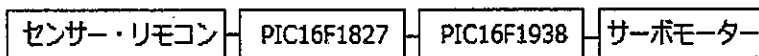
使うもの：圧電振動ジャイロモジュール『AE-GYRO-SMD』（秋月）

傾いた加速度を検知します。ここで異常値を検知すると、ロボットはまず倒れないようにバランスを取り

ます。これで倒れかかっても倒れないってわけですね。

Secret 7. The Heart

<ブロック図>



<説明>

基盤のサイズの都合で、40pinPICを使わずに 18pin+28pin です。18pin の 16F1827 で各センサー、無線リモコンを制御し、28pin の 16F1938 でサーボを制御します。16F1938 の方はほとんどの pin をサーボに使っています。基本的に 16F1827 で情報を処理し、サーボを動かす命令だけを 16F1938 に送ります。2 つはプリキユアなので心で通信します。嘘です。SPI 通信をしています

<回路図>

機密情報でバラしてしまうと国際警察に情報を渡すことになり、世界征服が遠のいてしまうので書きません。

<モーター>

サーボモーター(ブチロボ(Microsoft の 4 足)のサーボモータ)

今回のサーボモーターはすべてこれです。トルクが小さいですが、小さく軽く小型化にもってこいです。将来は、100t ぐらいのモーターで作る予定です。秋月にあるものはトルクが大きいですが重いです (Mogix(初代)に使用)。

Secret 8.

なんだかんだね、楽しかったよ。二足歩行を作ることは中 3 の時からだったけどやっぱり新鮮だった。今オレは“作っている”の感覚が常にあって (要はドヤ顔)、誰もやったことのないことをやっている (?、やろうと思えばみんな出来るよ) って感じでこれぞ制作、engineer って感じ。まあ残り残した (出来なかった) ことはいくつかあるけど (青歯、音声認識)、それでも、ここまでいったのは十分すぎると思う。

でも、もうちょっと頑張れた気もするね。春休み入るまではほとんど遊んでいたしね。でも、全ては無駄じゃないし悔やまれるものはないね。

最後に、もちろん、Mogix X は架空のものだし。最初の話もフィクション。でも、Mogix で世界を駆け巡るのは夢であることに変わりない。だれか作ってくれ。世界の夢を。

蜘蛛の眼～「登」に魂を注いだ者達～

製作:H2 栗本郁也、H1 金子真太朗、M3 朽木良輔

協力:物理部無線班の皆様、OBの皆様

i)Story

～アポロ計画やヒトゲノム計画といった人類の歴史に刻まれるような計画。

そんな計画の壮大さとは乖離した、この世界に五万とあるような小さな計画。

これは、そんな数多の計画の中の一つ、三人の「K」による計画。

その長い計画のたった一日、しかしこれ語らずしてこの計画を語る事は出来ない、ある一日の話である...～

"bee! bee! emergency! emergency!"

K「吸盤が限界だッ！」 K「真空ポンプも危ないッ！」 K「くっ!...、こいつももう限界なのか...」

K「そんなッ!蜘蛛の眼は俺らの長い夢の結晶なんだッ!締められるかッ！」 K「モーターが危ないぞッ！」

K「こいつも長い間頑張ったんだ...、そろそろ眠らせてやろう...」

～そう、この蜘蛛の眼は長い計画が故、その機体の限界を迎えていたのだ～

K「そんなのこいつも望んじやないッ！」 K「おいッ!モーターが限界だッ！」 K「くっ!...」

K「もう無理だ...、こいつがこれ以上登ることが出来る力は皆無だ...」

～蜘蛛の眼は、三人の夢を背負って登り続けているのだ～

K「頼むッ!もう少し待ってくれッ！」 K「これ以上、苦しんでまで見届ける意味など存在しないというのにな...」

K「...」 K「...、ッ！」 K「.....、ッ!？」

K「何故...、落ちない...」 K「モーターも吸盤も、ポンプも、もう限界を超えている筈だぞッ！」

K「これが...、奇跡、なのか...」 K「...」 K「.....」 K「.....」

K「何故だッ!、永遠など所せず最期を迎えたお前がッ!、何故ッ!、その身で登り続ける事が出来るのだッ!

否ッ!、命も魂も宿らぬその身でッ!、お前は何を望むのだッ! 蜘蛛の眼ッ!」

～三人の怨いを背負った蜘蛛の眼、その身の限界を超えて尚、使命を全うせんとする。

条理を超えたその身に、魂は宿っていたのかも知れない.....～

END

(作：厨二病が終わった筈の高一)

ii)概要

「蜘蛛の眼」は、壁に張り付いて進む事を目的としたロボットです。

「蜘蛛の眼」は現在二種類製作されているので、それらをそれぞれ「typeA」「typeB」とします。尚、どちらも、壁に張り付く手段として真空ポンプ(→説明は後述)を使用している点、魂が籠っている点に変わりはありません。

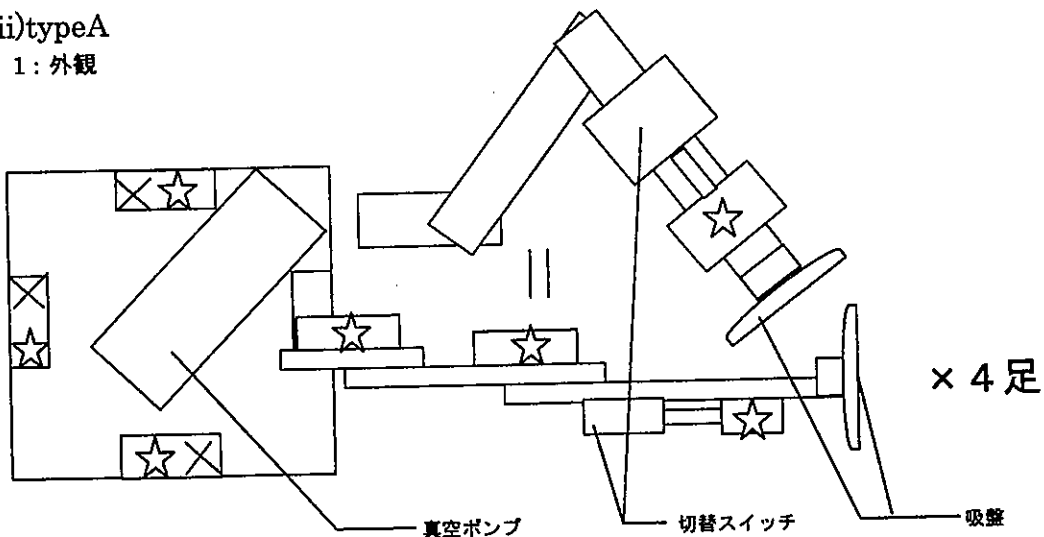
まず、「typeA」は四足の形を取っており、張り付きながら四足歩行で進みます。そのため、動力にはサーボモーターを使用しています。製作を担当しているのは栗本さんと金子です。

「typeB」には足は無く、タイヤが付いています。張り付きながらそのタイヤを回転させる事で進む事が出来ます。そのため、動力には DC モーターを使用しています。こちらの製作の担当は栗本さんと朽木です。

この二つのロボットを順番の解説していきたいと思います。

iii)typeA

1: 外観



☆ = サーボモーター

× = 本体側のサーボと足の接続部

※残りの足やポンプの配管、基盤などのその他細かい部品は書き記すと混ざって見づらくなるので省略しています

大分簡略化されていますが、これが本体です。図では省略しましたが、真空ポンプから切替スイッチに、切替スイッチから吸盤に、プラスチック製の管が配管されています。本体はアルミ板を使用しており、基盤はアルミ板を挟んで真空ポンプの反対面に取り付けています。

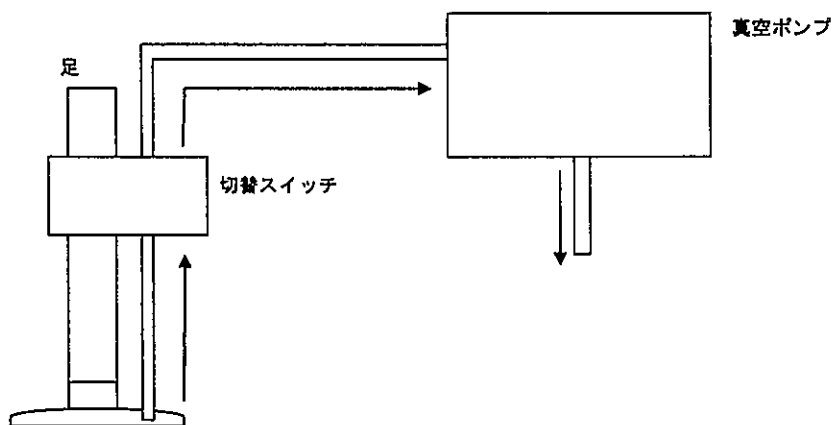
この図だけでは分からない事が多いので、各種部品を順番に解説していきます。

2: サーボモーター

このロボットには合計12個のサーボモーターが使われています。内8個は足を曲げて動かす為、残りの四個は切替スイッチ(→説明は後述)を動かす為に使われます。サーボモーターとは、何周も回るDCモーターと違って、角度を決めて180度以内で回るモーターです。サーボモーターの制御にはパルス波という波形が使われるのですが、詳しい説明は僕の役目では無いので割愛させていただきます。因みに、サーボモーターの制御にはPIC16F1939を使用しています。

3: 真空ポンプ

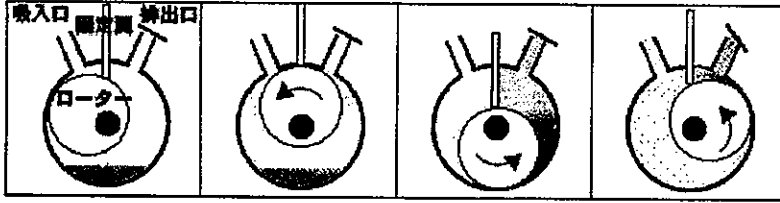
このロボットには、壁に張り付く為に真空ポンプという物が使われています。真空ポンプ自体の詳しい仕組みを説明する前に、まずは真空ポンプを使ってどのように張り付いているのかを説明していきます。



簡単に表すとこのような仕組みになっています。矢印が空気の流れの方向です。上図の時には切替スイッチは「開」になっており、空気が流れる事が出来ます。その為、真空ポンプによって吸盤の中の空気が吸われ、張り付く事が出来るという訳です。よって、切替スイッチを「閉」にすれば空気が流れられなくなり、吸盤が張り付かなくなります。

～真空ポンプの仕組み～

「真空ポンプ」と言っても色々な種類があり、今回はその中の一つ、「ロータリーポンプ」という物を使っています。

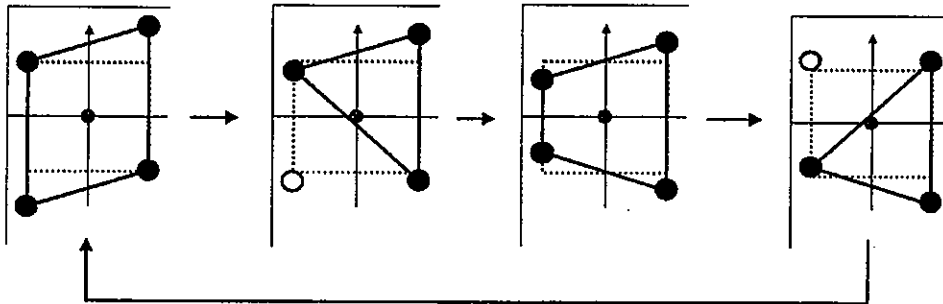


上図はロータリーポンプの仕組みです。上図のように、ローターは周りの壁と接しながら回転し、同時に固定翼も下がります。ローターが油面に入ると空気の逃げ場がなくなり圧縮され、一定の気圧より高くなると排出口から空気が出ていきます。こうして一つの口から入った空気がもう一つの口から出せるのです。

4: 歩き方

最後に、このロボットが壁に貼り付いた状態で、如何にして四足歩行で進むのか、その歩行パターンを紹介し、解説します。

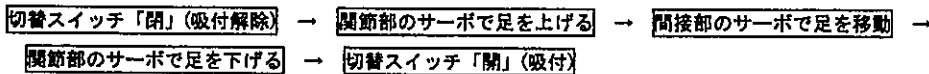
今回使っている歩行パターンは「クロール」と呼ばれる物で、常に足を三本以上着地させているので、自重をポンプの吸い付きだけで支えなくてはならないこのロボットに向いています。クロールは四足歩行の最も基本となる歩行パターンであり、爬虫類や哺乳類もこのクロールで歩行しています。



左右対称になってループ

クロールの歩行パターンは上図の通りです。黒丸は着地している足、白丸は浮いている足、真ん中の×点は重心を表しています。このループを繰り返す事で前に進む事が出来る訳です。

歩行時のサーボモーターの動きを図で示しておきます。



この動作が四本の足それぞれで行われます。

以上、「typeA」の解説でした。

iv)typeB

1: コンセプト

僕たちのコンセプトが「壁を登る」ことなので、それを主軸にすえています。僕の製作物については、本体を車輪で動かします。

2: 仕組み

機体の仕組みとしては、車体を真空ポンプで制御した吸盤で壁面に固定し、DCモーターを使った二つの車輪で壁面を登らせます。なお真空ポンプとDCモーターの制御には、両方PICを用いています。使っているPICは16F1939です。

～DCモーターについて～

DCモーターは二個の端子からそれぞれ、+と-が入ったときに正転、-と+が入った時に反転、-と-が入ったときに停止します。なので、ふたつのDCモーターの正転と反転を操る事で、前進、後退、方向転換を行います。

また、回転数の増減については、タイマー割り込みという仕組みを使います。タイマー割り込みというのは、一定してHの波形に対して一定間隔で短いLの波形を入れることで、全体におけるHの量を減らす事で、強弱をつけます。

～真空ポンプについて～

基本的に真空ポンプはDCモーターを使うので、上の文章を参考にいただければ幸いです

v)感想

金子：とりあえず、ここまで頑張ってきました。そろそろ完動させたいですね。

「壁を登る」のは想像以上に楽しいものでした。この製作物にして良かったと思っています。とは言うものの、やはり技術力があまりにも欠如していました。欠如してたのは脳味噌かも知れませんが、去年の製作でサーボをかじっていた筈なのに一からやらなければならなかったのは非常に良くないですね。

栗本さんには本当に御迷惑を御かけしました、すいません。色々と指導して下さい本当にありがとうございました。朽木も、俺が教えられる事が少なくて、自分でも先輩としてどうなのかと思っています。ごめん、そしてありがとう。お二人方共、これからの残された製作時間、宜しく願います。

☆カメラ部分

このロボットにはカメラモジュールを搭載して、ロボットからみた風景を PC に映し出せるようにしています。

使うマイコンは LPC1769 です。USB 仮想シリアルデバイスとして使用しています。

OOV7670

aitendo カメラモジュールです。電源電圧は 3.3V で、CLK は 25Mhz の水晶発振器を使用しています。ピンの機能は以下のとおりです。

XCLK	: クロックの入力です。10Mhz から 48Mhz で動作します。こちらの環境では 50Mhz でも動作しました。
PCLK	: カメラから 8bit のデータを一回送信する時に出力が変化します。
VSYNC	: 1 枚の画像を出力したら出力が変化します。
HREF	: 一列の画像を出力したら出力が変化します。
D0~D7	: PCLK に合わせて 8bit のデータを出力します。
VCC&GND	: モジュールの場合は 3.3V を入れれば動きます。
SIO_C	: カメラモジュールの設定を変更するための通信ピンです。こちらはそのクロックピン。
SIO_D	: カメラモジュールの設定を変更するための通信ピンです。こちらはそのデータピン。

OSCCB 通信

OV7670 はデフォルトだと美しい色を出力してくれません。色のバランスも崩れてしまいます。

そこでこの I2C に似た通信によって内部のレジスタに値を設定して、正しい画像が出力するようにします。

10 μ s のウェイトを信号に与えています。詳しいプログラムは Web にアップするのでそちらを御覧ください。

```
#define SCCB_DELAY 10
void sccb_send(uint8_t address, uint8_t data){
    signed int cnt = 0;
    sccb_start();
    sccb_ack();
    for(cnt=7; cnt>=0; cnt--){
        if((address & (1<<cnt))>0) SDA_H;
        else SDA_L;
        delay_us(SCCB_DELAY);
        SDL_H;
        delay_us(SCCB_DELAY);
        SDL_L;
        delay_us(SCCB_DELAY);
    }
    sccb_ack();
    for(cnt=7; cnt>=0; cnt--){
        if((data & (1<<cnt))>0) SDA_H;
        else SDA_L;
        delay_us(SCCB_DELAY);
        SDL_H;
        delay_us(SCCB_DELAY);
        SDL_L;
        delay_us(SCCB_DELAY);
    }
    sccb_ack();
    sccb_stop();
}
```

○画像フォーマット

画像のサイズは最大 640*480 です。SCCB により 320*240 などに変更可能です。今回は 120*120 程度で読み込んでいます。また、出力の色のフォーマットを変更することができます。今回は RGB444 を使用しています。

RGB444 : Red,Green,Blue のデータを 4bit ずつ系 12bit を 8bit のデータ送信 2 回によって送ります。

pixel ごとに 2PCLK が必要になります。

他のフォーマット : RGB565 や RGB555、Yuv の出力もできるようになっています。

左上からピクセルごとに色データを送り、1 列で HREF が、1 枚の画像で VSYNC が変化します。

○LPC1769 について

ボード	:	LPCXpresso	開発環境	:	LPCXpresso
動作周波数	:	100Mhz	使用機能	:	GPIO,DMA,USB Device

LPCXpresso が手軽だったので使わせて頂きました。STM32 の開発に使っていた某開発環境がコードサイズに制限が出来てしまったため、こちらにしました。

USB デバイス機能しか使っていないので、おそらく Cortex-M3 である必要はないのですが、積極的に新しいマイコンを使っていくことで拡張性を持たせたかったため使用しました。

○USB Device

最近の PC にはシリアルポートがついていないことが多く、USB が開発に使われるようになってきたと思います。

しかし USB-シリアル変換ケーブルや IC を使ったものでは通信速度に限界があり、デバッグ用としては十分なものの画像などのデータ通信には不向きと考えました。そのため USBDevice 機能を持ったマイコンを使用しています。

しかし、LPC1769 の USB 機能では 12Mbps が限界の Full-Speed しか使えないので High-Speed の物を使うべきだと思います。

汎用デバイスのドライバを作成する能力がなかったため、CDC のドライバをネットから利用させて頂きました。

○その他使用したツール

Visual Studio C++	:	PC 側のソフトの作成に使用しました。
Tera Term	:	USBDevice のデバッグで使いました。マイコンから printf() で文字を送ります。
315Mhz 無線モジュール	:	aitendo で購入しました。アンテナを使わないと 2m くらいが限界でした。
Xbee	:	データ通信の無線化に使用しようと思いましたが、速度不足。
STM32F4	:	フリーの IDE が使えなくなったので断念
ATmega328P,ATmega1284P	:	当初カメラのデバッグに使用。C コンパイラが使いやすいです。
128*128 液晶	:	BMP は表示できたのですが、カメラと組み合わせるのが間に合わず。
Altera CPLD	:	カメラのデータを 8bit から 32bit に変換しようと思いましたが間に合わず。

○現在の状況

画像の表示はできるのですが、なぜか赤色と青色が混ざります。緑色だけなら綺麗に映ります。プログラムの効率が悪いんだと思うんですが、120*120より画像を大きくすると動画ではなくなってしまいます。

○参考文献

過去の回路図集	: 参考になりました。バクリになりかけて申し訳ありません。
トランジスタ技術	: 素晴らしいです。本当に参考になりました。僕のやろうとしたことが出ます。逆では無いですきっと。
InterFace	: 付録ボードが毎回すごいですね。しかし、これを使ったら負けかなと思っています。
ネットの皆様	: 沢山の方々の文章やプログラムを参考にさせて頂きました。ありがとうございます。
データシート	: 間違ってるのはどうかと思います。皆さん注意しましょう。
試して学ぶ AVR 入門	: 種類は違いますがカメラのページが参考になりました。VC++も参考になりました。
Mbed の Cookbook	: mbed は使わないと心に決めていたのですが、プログラムは読ませて頂きました

○感想

栗本:

今回は4度目の共同制作でした。毎年全て共同製作です。これで悟ったことは共同制作はかなり難しいということです。よく起こる現象は“やる事が完全に分かれて互いに干渉しない”とか“どちらかに依存して片方がまるで努力せず教えられるだけ”とか“片方に押し付けて仕事が偏る”とかですかね。1つ目はまだしも2,3こ目はどうしても回避できないこともあるので、覚悟してやってください。どちらかが役職的に忙しいことも多いのでまあ仕方ないですね。不足分を補うための共同制作はいいと思いますが、2人とも暇なら一人で作ったほうがクオリティ上がりますよってのが結論です。まあやる気があって仲が良ければどうとでもなるよね。今年は楽しかったのでよかったです。2人ともお疲れ。

近年技術が進歩したと言われる物無ですが、モジュールを多用することで、独創性が損なわれてきたとはよく言われますね。ロボット系はみんなサーボ回してますし。こんな中一つ僕が思うのはコンピューター系とロボット系のつながりの薄さですね。独創性のある形のロボットを作るのも大切ですが、コンピューター系の技術と合わせると自由度は広がります。画像認識でモノをつかむとかイケメンですね。音声認識でお茶くみもいいと思います。コントローラーに液晶を入れて、ロボットの情報を表示するとかやりたかったです。そうした際のデバッグにPCを使うのは極めて有効だと思います。純正のお高いJTAGデバッガーを買ってもいいですが、FTDIのチップ使うなり、USBデバイスやるなりしたほうがクリエイティブでいいと思います。あとは誰かARM使ってほしいな…くらいですかね。以上です。

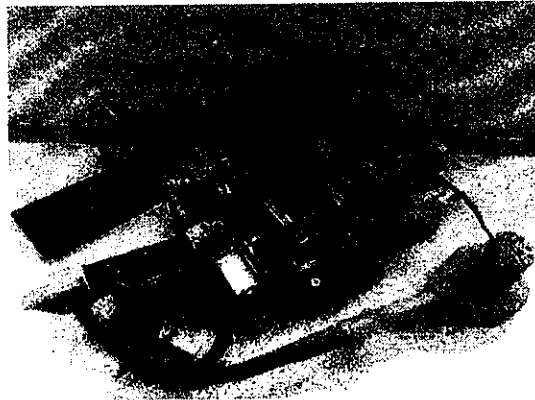
最後の制作にしては心残りもありますが、楽しかったし、そこそこ教えられたし、新しいこともできたと思うので満足です。強いというなら来年の2人の制作が心配ですね。応援してます。あああと、ブラウン管少年も頑張ってるね。

Alto

高1 河村

協力 物無の皆様

♪誕生物語



「音楽は人類に与えられた万国共通の言葉である」

Johann Sebastian Bach

音楽の成績が著しく悪かった僕。理由は小学生の時に遡ります。僕が小学校3年生までいた学校では、4年生から楽譜を習います。そして、転校した先の学校では、3年生までに楽譜を習うのでした。こうして生まれた音楽難民。リコーダーの試験では、カタカナをふらないとまともに演奏できない始末でした。僕は音楽とは縁がないのかな。

そう思っていた時期もありました。そんな時、救ってくれたのがこの「Alto」。僕は「Alto」を使って、音楽のテストでA+をとって友人を一泡吹かせてやりました。それからの人生は順風満帆そのもの。宝くじにもあたり、巨額の富を手に入れました。

最後にみなさんに、一言。変れるよ、現に僕は変わった。「Alto」で僕は変わったんだ。

♪概要

見てのままで、リコーダーを自動で演奏してしまおう！というロボットです。演奏方法は人間と同じで、空気を送るエアポンプと穴をふさぐモーターとで演奏しています。また、人間になるべく近い音を出すため、タンギングやサミング機能も実装しています。

♪機構

♪指

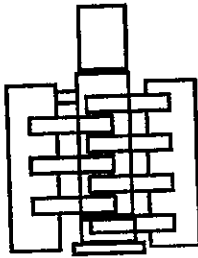


図 1

各指はこのように配置されています(図1)。指は、それぞれリコーダーの上側の穴を塞ぐために7個、裏の穴をふさぐ為に1個ずつついています。

それぞれが図2のように動くことで穴を塞いでいます。指部分の先端にある平べったいものは、耐震用につけるジェルで穴をぴったりと塞ぐためについています。ジェルは粘着性もあるため空気を密閉するのに非常に適しています。また2cm四方のものが105円と安価です

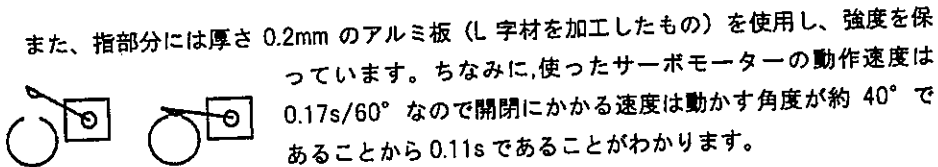


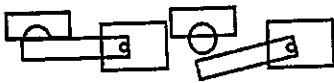
図 2

また、指部分には厚さ0.2mmのアルミ板(L字材を加工したもの)を使用し、強度を保

っています。ちなみに、使ったサーボモーターの動作速度は0.17s/60°なので開閉にかかる速度は動かす角度が約40°であることから0.11sであることがわかります。

♪サミング

サミングとは穴を半分塞ぐことで、高い音を出す技法のことです。ちなみに小学校の教育指導要領では、小学校の4年生で習うことになっています。サミングができないとアルトリコーダーでは吹ける音がだいぶ制限されてしまうのでサミング機能を実装しました。



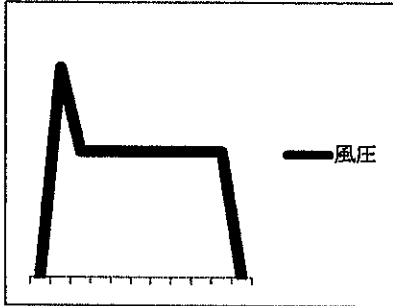
サミングは図3のようになっています。図3には半分開いているものと、全部閉じているものの2通りしかなく全部開けるものはありません。

図 3

しかし全部開けるのは高いソのみであること、またこのソでは空気を強く送ればソが出ることから、穴の半分を先にテープで塞いでしまい残りの半分を開閉することで、より綺麗で確実にサミングを行えるようになっています。

♪タンギング

タンギングとは音の出し始めに強く吹きトゥートゥーという感じで息を強く吹き下で空間を作ることにより、音の区切りをつける技法です。風圧と時間をあらかず関係は次のようになります。



このロボットでは、よりリアルな演奏をするためこのタンギングも実装しました。

しかし、使用しているモーターの細かい制御が難しいという欠点のためモーター単体でのタンギングは不可能でした。そこで、チューブをモーターによって折って空気を塞ぐ機械をポンプとリコーダーの間につけ、擬似的に口のような動きをさせることで、タンギングに近い動作を行いました。



このようにチューブを上から潰すことで空気を流れなくなるようさせています。

図4

また、モーターが入力値になるまで風圧が高くなるのに時間がかかること、音の吹き出しは強めにすることなどにより次のような4段階で一つの音を出しています。

- 1, ポンプを吹き始め用に強い風圧を送る (50ms 程度)
- 2, チューブを開く
- 3, 風圧を落とす。

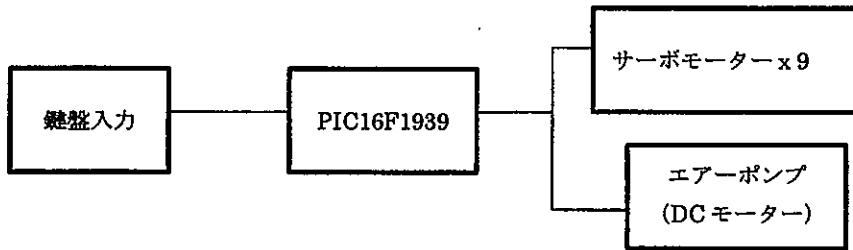
どの程度落とすかは音により異なります

基本的に低い音ほど送る風圧は弱くします。

- 4, 風圧を落とさずチューブを閉じて音を切る。

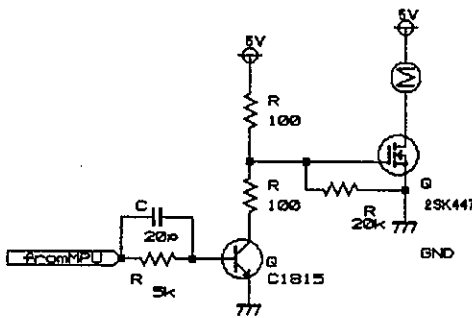
このように動作することでタンギングを行なっています。

♪ブロック図



マイコンは PIC16F193 を使用しています。またモータードライバ用の FET にはパワー MOS-FET 2SK477 を使用しています。

♪回路図



サーボモーターの回路は 16F1939 をつかっている点で他の人とかぶるので割愛しましたので、ここにモーター用の回路を載せました。片方にしか回転させないのでこのようにしています。FET は 2Sk447 を使っています。

♪プログラム

MPLAB IDE v8.76 と HI-TECH C Compiler を使いて書きました。プログラム自体は、複雑では無いので直接制御する部分を載せます。

それぞれの音に対応する指のデータは、次のように関数を使って書いています。power は風圧、singtime は音の長さです。

```
void G3(int singtime){
    servo[0]=1;
    servo[1]=1;
    servo[2]=1;
    servo[3]=1;
    servo[4]=1;
```



```
servo[5]=1;
servo[6]=0;
servo[7]=0;
power=1500;
```

```
}
```

また、楽譜などはこのようにして書いています。(チャルメラ)

```
C4(1);
D4(1);
E4(2);
D4(1);
C4(1);
delay_ms(500);
C4(1);
D4(1);
E4(1);
D4(1);
C4(1);
D4(4);
```

♪最後に

今年一年ほぼ一人制作（過去に 0 川とかいたような）でやって来ました。一人制作は初めてでした。一人制作は全部自分で調べたり、作ったり、はんだづけしたり、しなくちゃならないですが、そういうのもまた楽しかったです。反省としては、計画性が足りなかったかなとか思っています。まあ、なんだかんだで僕よりもうまくりコーダを吹けるロボットができたのは、よかったんじゃないでしょうか。充実した一年で満足です。



製作者/H1岸田聖生 M3佐藤惇 協力者/物無の皆様・ネットの皆様

◆ストーリー

某国紛争地帯――

「ハア……ハア……ハツ……ハ……ハア……」

俺はアサルトライフルを手に抱え、街を歩いていた。

右足からは血が流れ、ずきずきとした鈍痛を発している。

俺の名前は山本。

兵士としてこの国へやってきた。そして戦地へやってきて、今も戦場にいるのだった。作戦行動を行うべく移動していた俺たちの部隊の前に装甲車が突如として現れ、車両との戦闘などみじんも想定していなかった俺たちはんでバラバラ、俺も逃げる途中で敵兵に撃たれつつもからから逃れてきたのだった。

自らの立場を再確認しつつ歩いていると、左足に鋭い痛みが走る。

――どうやら足をくじいてしまったらしい。

足を引きずりつつ、建物にもたれかかり、やけに晴れ渡った空に目をやる。

その時、地面が揺れ始めた。地震のような振動は装甲車がこちらへきている証拠だった。

今――歩くことはできない。おそらくこのままでは隠れることだってできないだろう。

手持ちの武器は、アサルトライフルに拳銃一丁、それに手榴弾が一つ。

アサルトライフルや拳銃では到底装甲車を倒すことはできない。手榴弾を使えばかろうじて走行できないようにするぐらいならできるかもしれない。だが、この距離でそんなことをすれば自分の体ももちはしないだろう。

――ふと、頭に昔の友人の顔が思い浮かんだ。

浮かんで消えるその顔は皆、戦争が起きるなんて誰も考えなかったような時代の友達。中学、高校生活をともに送った友達。部活で出会い、話し、ふざけ、時に罵り合い、そして、ともに笑いあった仲間たち。

――皆いい友達たちだった。

一人は戦地へ行き、一人は将校になるべく勉強中。あるヤツは多脚戦車を作っているらしい。

「佐伯、石倉……宮嶋……渡辺……藤川……市川……戸田……横尾……」

かつての友たちの名を呼び、そして、呟く。

――小さな……ささやかな、遺言だ……どうか……君らに……幸せを――

装甲車が十分に近づいたのを音で感じ、ピンに手をかける。

その時、俺は不思議な音を聞いた。

「シュツ!!!」

空気を裂くような不思議な音とともに、装甲車が火花を散らし、次の瞬間動かなくなる。

俺は音のした方を見る。

道には大きな灰色のクモがいた。こいつには、思い当たりが…ある。

これは確か…アイツが作っていた戦車…

俺はその戦車に描かれた文字を見つめ、呟くのだった。

「T6…the tiny testament of the true time…」

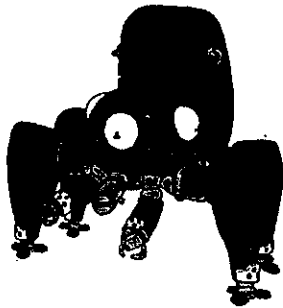
(T6…真なる時間の小さな遺言…)

ひどいストーリー詐欺です。ロボットの名前を除き、ストーリー内での出来事はロボットとも現実世界の人物とも一切関係ありません。

◆概要

このロボットは多脚戦車と呼ばれるものです。

この戦車のモチーフとしましては



↑コレや



↑コレ

などがあげられます。

二つでずいぶんイメージは違いますが、どちらも多脚戦車という意味でいえば同じジャンルの兵器になり、T6もこの分類となります。

多脚戦車という言葉自体が日常的に使われるものではないと思うので、ここで多脚戦車の説明しておきます。

現在、世界中の戦地で使われる戦車はたいていがクローラ(キヤタピラーは米キヤタピラー社の登録商標。)を履いた形状になっています。

ここでは多脚戦車のメリットとデメリットについてまとめました。

	通常戦車	多脚戦車
メリット	速度が速い 振動が比較的少ない。 比較的低い技術レベルで生産できる。 積載能力を増やしやすい。 コストが抑えられる。	障害物に対処しやすい ダメージに強い(一脚が破壊されても走行は可能) 機動性が高い。
デメリット	障害物に対する対処能力が低い。 ダメージに弱い(通常は二輪なのでクローラが片方でも壊れれば走行不可能になってしまう)	速度を上げづらく、振動も大きい。 高度な技術レベルを必要とする。 積載能力が低くなりがち。 コストがどうしても高くなる。 部品数が増え、不具合の原因となる。

いかがでしょうか。

こう見ると多脚戦車にはデメリットが目立つように感じられるかもしれませんが。

確かに現在の技術では多脚戦車の実用化は難しい現状があります。

しかし多脚戦車のデメリットはほとんどが技術レベルに起因するものとなっています。技術レベルを上げれば速度、振動、積載能力などのデメリットは解消されることとなります。

一方、通常戦車のメリットは根本的に解決が難しいものばかりです。

障害物に対する処理はサイズを大きくする以外に根本的に解決法がなく、兵器などにおいて大型化は致命的なマイナスとなります。ダメージ対策としましては迎撃兵器などを搭載することが解決策となりますが、これらの開発には高度技術が必要なばかりでなく、攻撃兵器の改良の後を追う形となり、根本的解決策は難しいものです。

このように、多脚戦車は将来性と発展の可能性を秘めた形でもあるのです。

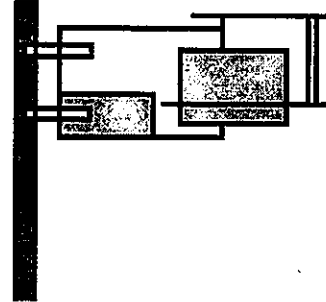
ただし、兵器としての実用性はあまり高くありません。

◆仕様詳細

関節自由度	13(各脚2×6+砲台稼働用上下)
サイズ	およそ40*40*15(cm)
重量	4Kg程
電源	5V4A
操作系統	有人

◆しくみ

ここではこのロボットの構造、しくみなどを解説します。
このロボットの脚の構造は大まかに右図のようになっています。構造自体は単純なものですね。
右の薄い灰色の長方形が水平回転用(足の横運動)、真ん中の薄い灰色の長方形が垂直回転用(足の縦運動)で、濃い灰色の部分を上下に動かします。これが六角形状に6本ついています。
ココから各部の説明をします。



➤ PIC16F1939

このロボットの頭脳です。ここにプログラムを書き込み
ここが計算をし、各部に指示を出します。
取り外すとこんな感じ(右図)のけじけじ状の IC になっています。



PICというのは IC のシリーズの名前で
「16F」中くらいの性能のシリーズの中の
「1939」=最新品番の中のピン数の多いモデル
ということを表してくれています。

➤ コントローラー

操縦者が触れる部分です。無線での通信を目標としています。

➤ サーボモーター(後述)

このロボットの筋肉、原動力です。

GWS 社製。

内部に制御 IC とモーター、センサが一緒に入っている
という優れたものでパワーもあり、優秀な部品です。なので高いです。壊しまくったせいで関節減りました。

PIC から受けた命令通りの角度に各関節を曲げることができます。

➤ コイルガン

戦車でいう主砲になる部分です。これがないと戦車とはいえなかつたりするわけですが、搭載できるかは未定です。原理を説明すると長くなるので割愛させていただきます。ガウス式加速器みたいなものだと思っていただいて結構です。

回路は PIC にコントロールの出力端子とモーターの入力端子がつながっているというだけの単純なものですので、省略とさせていただきます。



★サーボモータとは？

ざっくり説明してしまいます。

➤ そもそもサーボモータって？

ほかのロボットでも使用されているサーボモータ。そもそもこれはどのような物なのでしょうか？

サーボモータを wikipedia で調べると「サーボモータ(*Servo motor*)とは サーボ機構において位置、速度等を制御する用途に使用するモーターである。」と、あります。

これで大体意味はわかります。つまりサーボモータとは「位置、速度」などをデータとして与えることでその通りに動いてくれるモータのことを言うのです。サーボモータには普通のモータと異なり、回転に限界があり「＋－何度」といった範囲で稼働させます。

➤ サーボモータの構造

サーボモータは主に2つの部品から構成されています。ひとつは制御部分。もうひとつがモータです。制御部分は現在の軸の角度と外部から与えられたデータの差を求め、それを0に近づけるべくモータに電流を与えます。モータは普通のものと同じ電流を受けると力を発生させます。それらを組み合わせることで回転→制御→回転を繰り返し、命令通りの角度に軸をセットするようになります。

➤ 制御方法

サーボモータにはパルス波(電流を入れたり切ったりを高速で繰り返す)を与え、制御します。このようにパルスのうちの電流が流れる時間を増やすか減らすかによって制御します。一例として、パルス長が1msで－方向、2msで＋方向に振り切り、1.5msで中立態となります。

この時間は製品により異なりますが、大体のサーボは同じような方法で制御できます。

◆プログラム

これをPICに書き込み、計算をしてくれることでロボットが実際に動くこととなります。

プログラムとしては、脚の位置(事前に設定)を順番にサーボへと指示することで歩行を行います。サーボモータへの指示など、細かいことはここでは割愛させていただきます。余計なストーリーのせいで場所が少ないんです。興味のある方はインターネットで「サーボモータ 制御」などのワードで検索すると出てくるかと思います。

コイルガンの部分は物理スイッチ等のみで動かしています。

どのみち、小難しいことはしてないと言っていいでしょう。

◆現状(12/4/06)

ほとんど完成しました。やったね！

現在の改善点として、まず足の摩擦が少なすぎて滑ること。進みが遅いしたらありやしません。それに、右側の足が実は常に接地してない要らない子になりかけてるので左右に旋回する時の角度に差ができています。

また、砲身の上下のプログラムがものすごく適当です。砲身動かしてる間は本体は動けず、射角を維持したまま歩くこともできません。何が悪いかっていうと完全六脚歩行の時のプログラムを流用してるからですね。関数の書き換えとか大変で無理やり使い回しています。

まだ時間があるようなので簡易ライトレース機能でもつけたいかなーなんて思ってます。バッテリーは値段が高いのでスルー。そうすると必然的に無線リモコンである必要もなくなってしまいます。追加装備を考えるくらいでしょうね。

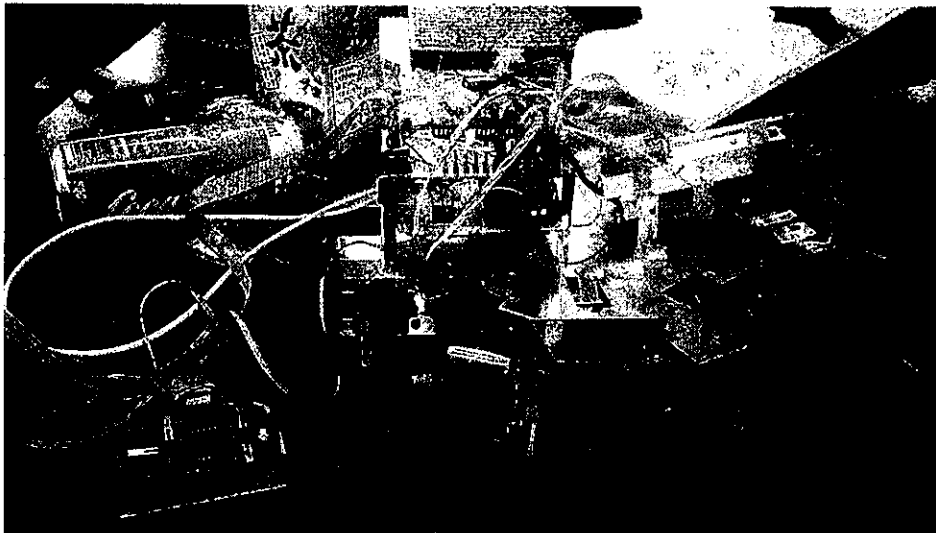
◆感想

岸田:ロボは動きが目に見えて面白いですね。サーボが動かなかった理由が謎です。

実質ハードウェアソフトウェアどっちも自分でやりました。もっと佐藤が作った部分増やしてやれたらよかった気がします。

佐藤:サーボも満足に動かせないまま売り物(TETOIS)に選ばれてしまったのでもはやこの制作物の頁に名前が載ってるのが不思議なくらいです。文責として:著作権といっても転載するほどのものもありませんね。実はプログラムの原型がコピーだったりするのでそっちのほうが心配。あと売り物の名前がすごくマズい気がしないでもないです。

↓現状



南風に靡く齒車

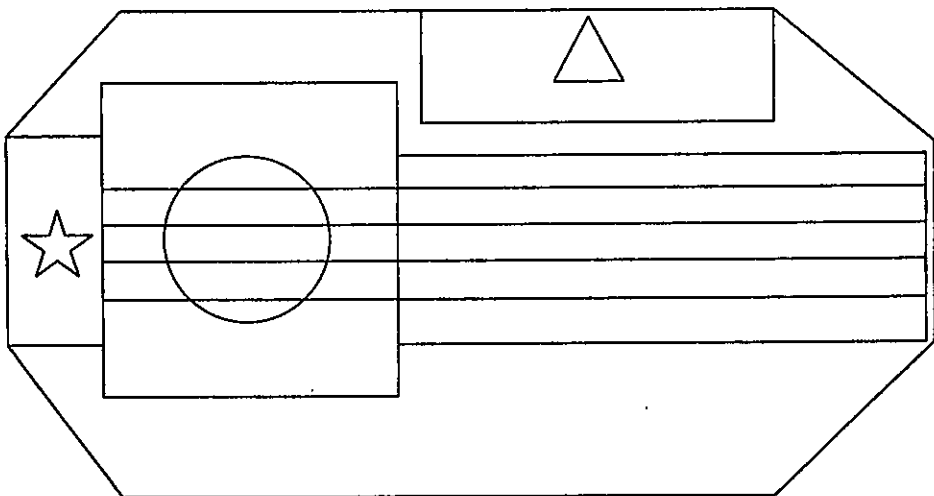
製作 M2花園佳月
協力者 物無の皆様

～STORY～

「どうすんだこれ!」「もう時間ねえぞ!」
怒声が飛び交うここは、とあるラジオ局。今日は、国民的ウクレレ奏者
T氏に来ていただき、演奏してもらおう予定だった。
予告までしたのに、何と当日T氏にドタキャンされてしまった!
誰もがこの事態に悩む中、一人の男が口を開いた。
「ロボットで代用しません?」
「そんな時間ねえよ・・・」「勝手にしろ」
という批判を押し切って、彼の孤独な戦いが始まったのだ。

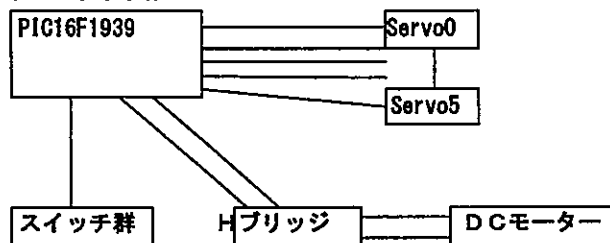
～概要～

要はウクレレ演奏ロボットです。
サーボモーターで弦を押さえ、DCモーターで弦を弾いて演奏
します。手動/自動演奏の切り替えができ、自動でデモ演奏ができます。
外観図下の通り↓



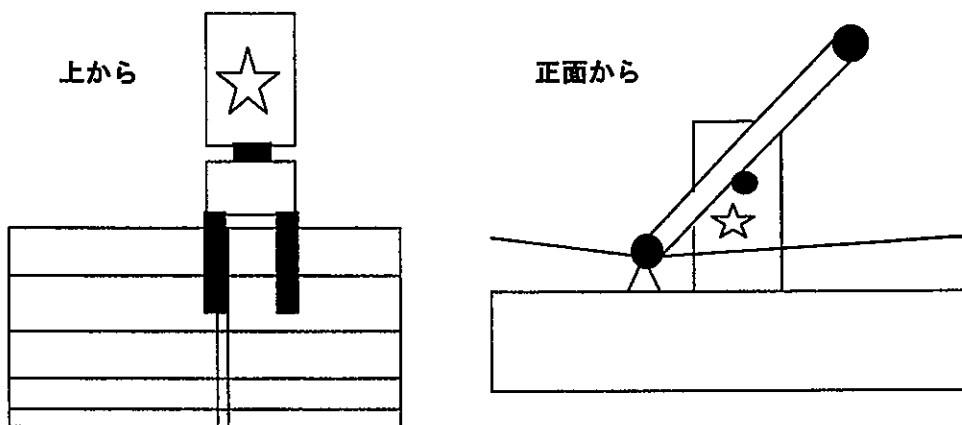
- 星印・・・DCモーター
- 三角印・・・サーボモーター群
- 中央・・・ウクレレ

・回路 (ブロック図)



・サーボモーターについて

サーボモーターとは、指定された角度に動くモーターです。
 +と-の入力線の他に、一本の信号線があり、そこに20ms間に
 どれだけの時間Hを出すかで角度を決めます。
 僕はサーボに棒を取り付けて、下図のように弦を押さえています。



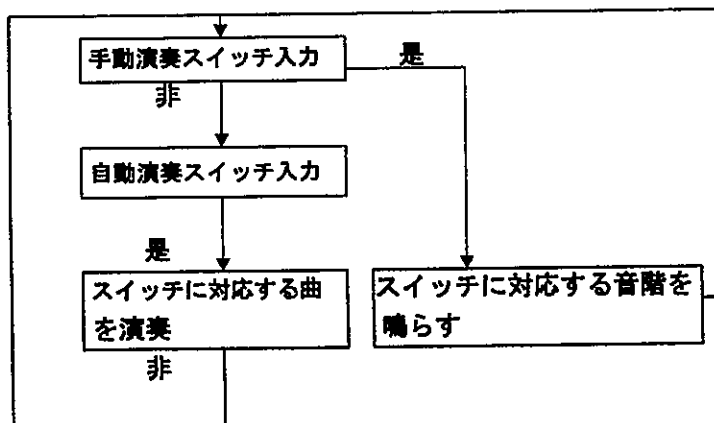
星印がサーボです。

・DCモーターについて

DCモーターなんて言い方がややこしいだけで、実際は誰もが小学校の理科の授業で使ったことがあろうあのモーターです。制御にはHブリッジという回路をしようしています。Hブリッジとは、FET というスイッチ的な物で正転/逆転を切り替えられる回路です。TA7291 を使ってます。

・プログラム

長いのでフローチャートを載せます。



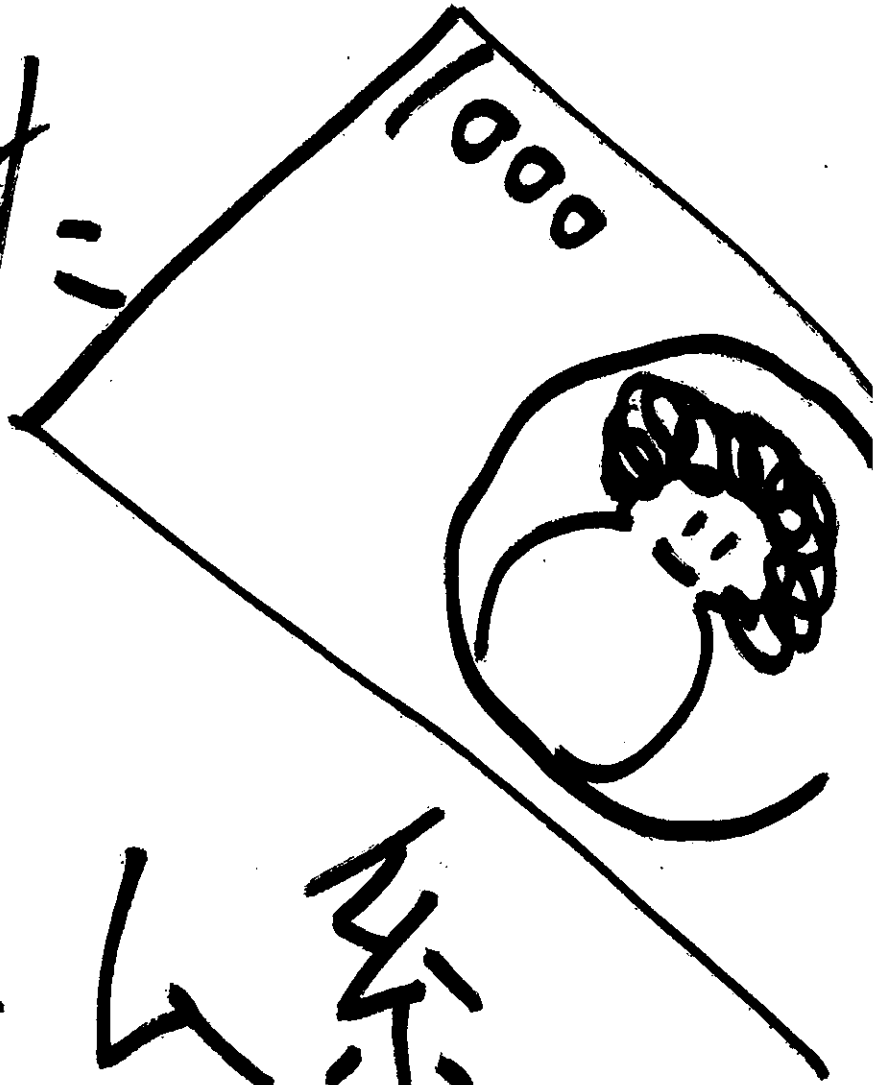
～感想～

C言語は、始め全く分からなかったのですが、そのうち少しずつ分かってきました。先輩方本当に有難うございました。

現状として、モーターの駆動音が五月蠅くてウクレレの音あまり聞こえません。さらに今まで、PICを5つも殺してしまなど数多の問題も生じていますが、文化祭までにはまとも物にしあげたいです。

野口は

見た見



ゲーム系

物無員トローゼ

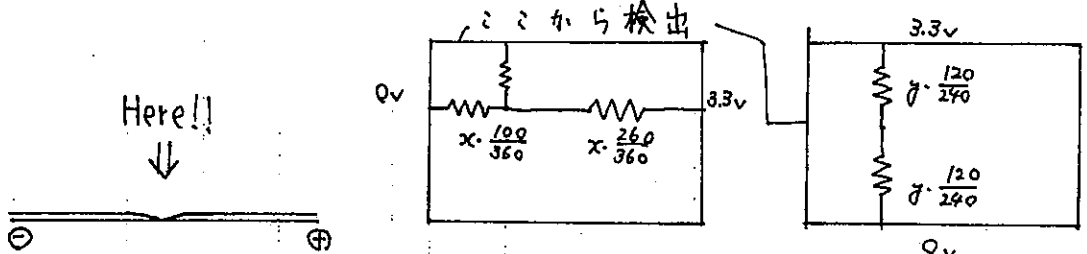
制作者 H2 石原 皓
M3 黒田 健太

～ゲーム説明～
単純に落ちて消えます。な
とて消えます。な
ズルズと珍
ゲームの効
ムれ音も
す。ラッパ
数種おる
アッパッ
イ動お
コンカ
が、の
54声
×4を録
4以上音
課題再生

～Story～
「な」とい
た彼の自信
が余るの
せよめ
るのか!?
ストーリー
だ、去った
「お前さん
親、うち
決、決、決
彼は決
果、先生は
「お前さん
決、決、決
た、三人揃
。余りには
30年消え
近すえい
教「!!」し
鞭四、う、
を人う、
と「の、
て、上、
き、童、
を、班、

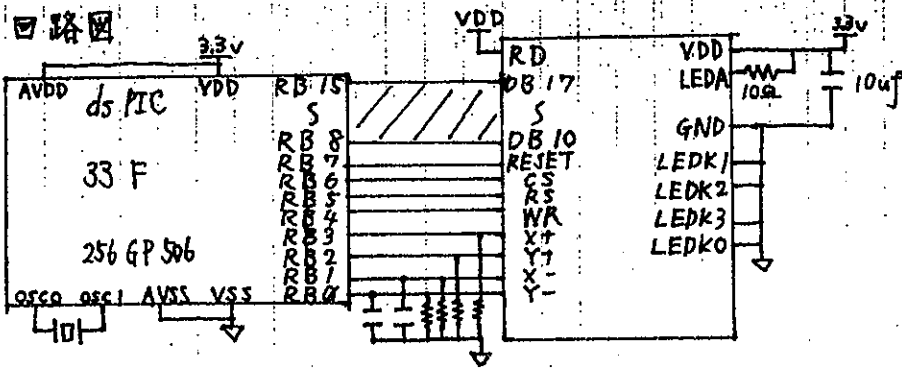
～液晶表示～
僕が担当した液晶表示について書こうと思います。
マイコンはPIC256GP506、液晶はHY024006Aです。
液晶の初期化、点表示、全体クリア、文字描画などの関数は、後で書かせていただきます。
液晶の初期化、点表示、全体クリア、文字描画などの関数は、後で書かせていただきます。

～タッ子用スクリュー～
今回使ったタッ子用スクリューは、液晶に接触する部分に、抵抗を付けた構造で、目子軸と子軸の順です。
タッ子用スクリューは、液晶に接触する部分に、抵抗を付けた構造で、目子軸と子軸の順です。



x軸 ⇒ 子軸の順です。

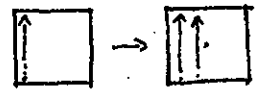
回路図



〜プログラム〜
アイコン表示

```
void lcd_Icon(int a, int b, int icon type){
    int x, y;
```

```
    for(x=0; x<32; x++){
        for(y=0; y<32; y++){
            lcd_Pixel(32*btty, 32*atx + x, Icons[icon type][y][30-x]);
        }
    }
```

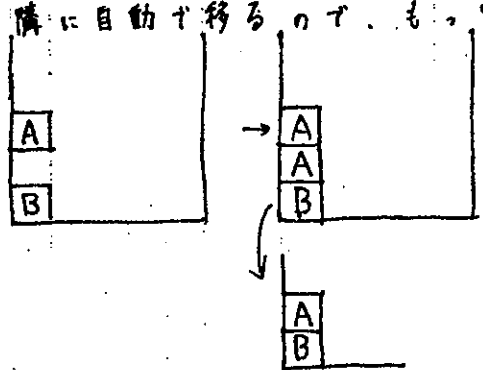


※図はイメージです

↓ 実際は1ピクセル表示するごとに隣に自動で移るので、もっと効率を良くすることもできます。

スクロール

```
void lcd_Scroll(int a, int b){
    int a2, b2;
    if((Zahyo[a][b-1]==0)&&(b1==0))
        lcd_Icon(a, b-1, Zahyo[a][b]);
    Zahyo[a][b-1] = Zahyo[a][b];
    Zahyo[a][b] = 0;
```



↓ まず表示を移した後は種類の判定を移動しています。再帰処理により、さき
個数判定 Type[a][b] = Zahyo[a][b]; しました。同じ処理を繰り返させています。
void Troze(int a, int b){ kosuutt; Zahyo[a][b]=0; 左の計文のZahyoはType関数です。

```
if(Zahyo[a][b] == Zahyo[a+1][b]) Troze(a+1, b);
if(Zahyo[a][b] == Zahyo[a-1][b]) Troze(a-1, b);
if(Zahyo[a][b] == Zahyo[a][b+1]) Troze(a, b+1);
if(Zahyo[a][b] == Zahyo[a][b-1]) Troze(a, b-1);
if(kosu < 4) Zahyo[a][b] = Type[a][b];
```

〜感想〜

あまり製作にはかわれず、石原さんに多大な迷惑をかけました。本当に申し訳ないです。なんとかプログラムを理解するところまで行けました。

ここより石原がお送りします

○音について

まず、音とは何でしょうか？高校物理では一般的に「波動」の形態の1つとして教えます。

「波動」とはエネルギーを運ぶ手段の1つです。

波を伝える物質(媒質)はその場で振動するだけです。

媒質の振動とともに、エネルギーが伝わるのです。

音は縦波(=過密波)ですがグラフに表しにくいので横波として表示します。

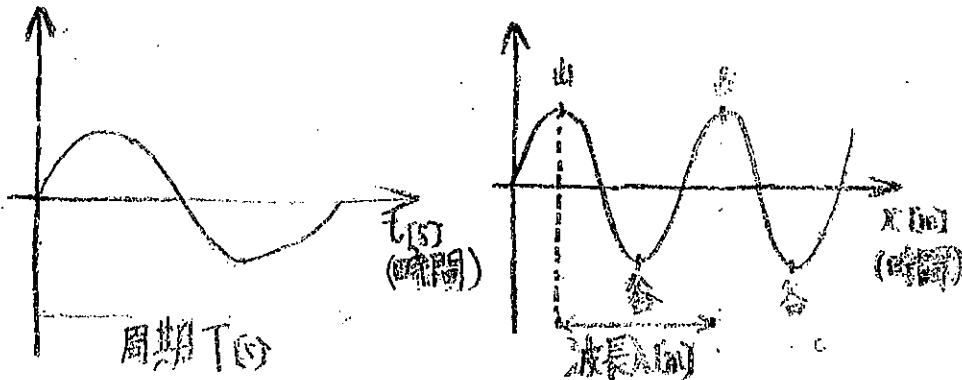
まあ、ドップラー効果等は扱わないため、縦波・横波の区別はしないで頂いて結構です。

☆波の基本式

v =波の進む早さ λ =波の波長 T =波の周期とおくと、

$$v[m/s] = \lambda [m] / T[s]$$

波の進む速さは問題無いでしょうが、波長と周期について軽く触れておくと、



周期は、媒質が1回振動するのにかかる時間

波長は、山の位置にいる媒質同士の間隔
となります。

☆振動数 f …1秒間に媒質が振動する回数。

$$f[Hz] = 1/T$$

波について語る上で重要なものを挙げてみました。

では、音についてももう少し深く見ていきましょう。

別に製作物に関係ない話はしてませんよ？

なんか真面目過ぎて物無の回路図集っぽくないなあ

○音の3要素

- 1、音の高さ 波の振動数が大きくなると音は高くなります。
- 2、音の強さ 波の振幅が大きくなると音は大きくなります。

エネルギー 媒質の密度 波の振幅 波の速さ

$$E[J] \propto A[s/m] \times (B[m])^2 \times V[m/s]$$

- 3、音の音色 波の波形が違くと音色の違いが生じる。

1、音の高さについて

音程の取り方には大きく分けて2つあります。

(1)ピュタゴラス律

ピュタゴラス、というと数学や天文学の印象が強く、音楽の話に出てくる事が意外に思われるかもしれませんが、近代に入るまで学問には哲学・神学と自然科学の2つしかありませんでした。彼が琴のような楽器で実験をしたという話です。

彼は和音の特に美しい倍音などについて研究していたようで、「なるべく簡単な整数比で表せる周波数を持つ音の和音は美しい」ということを知っていました。そこで、半音7つ分離れた2つの音の周波数比を2:3、半音12つ分離れた2つの音の周波数比を1:2として、他の音との周波数比を求めたのです。

(2)平均律

ところがピュタゴラス律には大きな問題点がありました。1つの決まった音と残りの音との和音は美しいのですが、残りの音同士ではきれいな和音にならないため、そのメリットが生かされる場面が非常に少ないのです。例えば小学校で誰もが教わる和音として「ドミソ」がありますが、ピュタゴラス律に合わせた楽器では「ミ」の音が浮いてしまいます。この解決案としての妥協の産物が「平均律」です。

半音ごとの振動数の比を $^{12}\sqrt{2}:1$ としています。現在ではこちらが主流です。

この製作物では計算の簡便さから(1)と(2)を使い分けています。

なお、基準音(ハ長調)の周波数は440~446Hz周辺に設定されます。

オーケストラなどでは若干高めに444Hz周辺にする事も多いですが基本は440Hz

です。この製作物では444.44...Hzを用いています。

2、音の強さについて

波の振幅とは媒質のふれ幅をいいます。

3、音の音色について

「ア」と「イ」が違って聞こえるのは何故か？コンデンサマイク(後述)を通した音の波形をオシロスコープで見ると周期は同じでも形の全く違う波が表示されます。

これが音色の違いなのです。

さて、やっと導入が終わったといった感じですね。導入が分かれば後は簡単です。

○録音再生IC

コンデンサマイクからの入力を内部でAD変換(きっと山本の説明があるでしょう)して保存、それをDA変換(AD変換の逆)して波形を出力しています。部員の断末魔の発生源。ぶっちゃけキットからパクったものなのでデータシートが以上に親切。

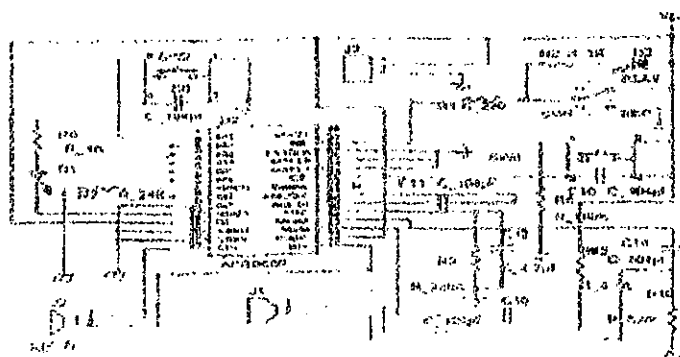
(回路図)

/M1_Message	1	28	VCCD
/M2_Next	2	27	/RE
/M3	3	26	EXTCLK
/M4	4	25	MSEL2
/M5	5	24	MSEL1
/MG	6	23	/CE
OscR	7	22	/Strobe
/M7_END	8	21	Ana_Out
/M8_Option	9	20	Ana_In
/Busy	10	19	AGC
BE	11	18	MicRef
VSSD	12	17	MicIn
VSSA	13	16	VCCA
SP+	14	15	SP-

○コンデンサマイク

直流に対しては電流を貯めて放出し交流に対しては抵抗として働く性質を利用したもの。声によって振動板を振動させ、コンデンサの容量を変化させる事で、電圧を変化させています。

(回路図)



○PIC(16F886)

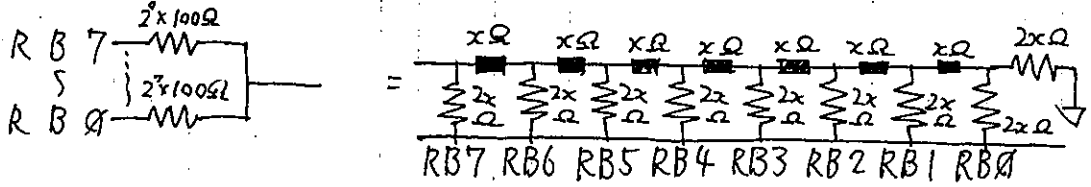
PWM(ロボットの誰かが説明するでしょう)を出力して音としたり、8bitでDAコンバータを制御したりSDカードと通信したりと忙しいヤツです。さらにまだPIC33FJ256GP506(液晶用)との通信という大仕事や録音再生用ICの制御という役割も。なお、(2012 2/22現在)SDカードとの通信は動いていない。

○DAコンバータ

PICの8bitのデジタル出力をアナログの波形に変形する。波形を変えられる、音色担当。

ラダー抵抗

D/Aコンバータをこれで代用していた頃もありました。
各ピンから出る電流の比を抵抗で制御して合計を出力します。

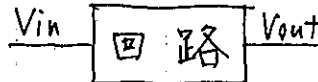


RB0 ~ RB7において、5Vを出力したらVDD接続と同値、0Vを出力したらGND接続と同値と考えると考えやすいかもしれません。

★ラダー抵抗を考える上で重要な法則

・オームの法則 V (電圧) = I (電流) R (抵抗)

・キルヒホッフの法則



$V_{in} = V_{out}$

↑当たり前に見えてすごく見落しやすいです

ラダー抵抗は出力する電流がとてもなく少なくなってしまふこと、抵抗値の誤差によるずれがあることが問題点。

↑ 金属皮膜などである程度は改善可能?

感想・反省

- ・今年で最後だから...と挑戦した音。なぜかみんなに破らされました。
- ・時間があたら加工にも挑戦してオルゴールとか作ってみたか、たです。
- ・2年目の共同製作。もう少し液晶を見ておれば良かったかな。

~ 少しでも遺言に書きそびれたことを ~

みなさんも共同製作をしたリ中リを見ることか必ずあるでしょう。そのとき以後輩が問題のある態度をと、たしします。みなさんはどう対処しますか？このとき、共同製作者は相手に対して手を差し述べてあげて下さい。自分が悪かろうが、相手は確実に苦しんでいます。相手の存在を責方が否定しては、誰も相手のことを見っめ、一緒に考え、ともに成長することはできないのです。共同製作者というのには他にいない存在です。

at 美工. 2012. 4/25 16:30

SPACE STONE SHOOTER

製作者:H1 山本 涼一

:M3 四柳 雄太

協力者:物無の皆様

1、はじめに

石、それは古来より人類の発展とともにあった。石器、建材、燃料、嗜好品・・・など中でも武器としての石の発展はめざましいものがある。日本でも古来より投石は狩りや遊びだけでなく、戦乱にも活用されていた。そのような武器としての投石を「印地」と呼び、印地をするものを「印地使いと呼んでいた。印地は時には弓より長い射程を飛び、防具の上からでも衝撃を与えたといわれている。戦乱で印地部隊が存在したほどであった。そして、印地使いの中で究極的に印地を極めた者は「印地打ちと呼ばれていた。伝説の印地打ちはこのような言葉を残している。

“印地打ちの真の武器はその強き「意思」なのだ。”

2、ストーリー

佐藤博はこの言葉が好きだった。

この一言で誇り高い印地打ちの精神をよく表している。

こういう風に生きたいと思っていた。

いや、こういう風に生きたかった。

...

さかのぼること三か月前。

佐藤家はその日も何事もなく平穏に暮らしていた。

強く、頼れるお父さん。誰にでも優しいお母さん。かわいい弟。

そのすべてが一瞬にして奪われた。

この地区を支配している恐怖の四天王のひとり、四柳のしわざだ。

四柳は目障りな印地打ちを殲滅するため、僕のお父さんにあらぬ容疑をかけ、僕の家を突如襲ったのだった。僕はその時運よく印地打ちの道場に行っていたためなんとか生き延びることが出来た。その日から僕は印地打ちとして生き、復讐をすることを誓ったのだった。

苦難の修行の末に僕は四天王の四柳、三柳、二柳、一柳を倒した。

そして、最後の敵、魔王「零柳」が今日の前にいる。

「ふっふっふ、石がなければ印地打ちもただの人だな。貴様はこの石のない世界“フラットゾーン”で私に勝つことは絶対にできないのだ」

ああ、もう手を動かすことすらできない。

手持ちの石はすべて使い切ってしまった。もう零柳を攻撃する手段はない。

僕はこれで終わりだ。ごめん、お父さん、お母さん、弟。仇をとることはできな

”印地打ちの真の武器はその強き「意思」なのだ。”

…そうだ僕は印地打ちだ。石がなくても自分の意思で戦える。

「ほう、まだ立ち上がるのか。そんなに私に殺してほしいのかッ！」

「僕は印地打ちだ。僕は自分の意思で戦う。自分の石でお前を殺すッ！」

その時、僕の手の中には確かに「石」があった。

3、ゲームの概要

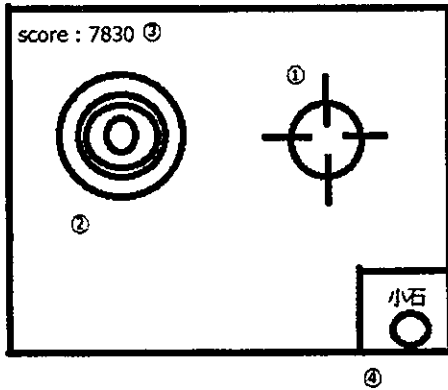
このゲームは的や敵を撃ついわゆるシューティングゲームです。

主人公:佐藤偉。鈴木だったかもしれない。現代によみがえった印地使い。究極の印地打ちを目指す14歳の少年。家族を殺された恨みで魔王零柳を倒すことを誓う。

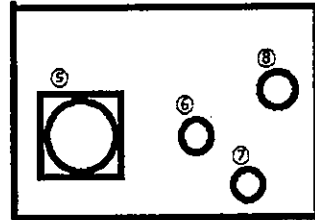
はい。これより上のくそどうでもいい設定まで目を通していただき、ありがとうございます。印地打ちってなんだよ。これから下はまだ真面目なのでそろそろ飽きてきた方も読んでいただけるとありがたいです。

ステージを選択しそのステージ内でのスコアを競います。ステージ内では様々な的や敵が出てくるので三種の石で倒してください。一つのステージでは出てくる的や敵が決まっています。

ゲーム画面



コントローラ



- 1 カーソル:ここらへんから弾が発射されます。
- 2 的:弾を当てると点数が入ります。
- 3 得点:多いとうれしいし、少ないと悲しいやつです。
- 4 弾種:今撃っている弾の種類です。
- 5 カーソル移動/選択:カーソルを移動させるコントローラーです。または選択。
- 6 発射/決定:弾を撃つボタンです。または決定。
- 7 弾種変更/戻る:使う武器を変更します。または前面面に戻ります。
- 8 ポーズ:一時停止です。

また、上記の通り使う武器を状況に応じて変えることが出来ます。

礫:威力はないが、広範囲に攻撃可能。敵の飛び道具を撃ち落とせる。

小石:連投可能。威力は弱いが高精度が高い。

岩:投げるのに時間がかかる。爆発する。弾数制限あり。

・音響部分

M3 四天王ではない四柳

4、使った部品

・PIC32MX440F512H

MP3 デコーダと SD カードの間に入って通信・制御をする CPU。最高命令実行速度が 80MIPS、多くの内臓周辺モジュールが実装されているなど高性能なマイコン。

・VS1011e

VLSI ロジック社製のオーディオコーデック、MP3 を再生するのに有名な IC。CPU から送られてきたデータをデコード処理して出力してスピーカなどから音が出るようにしてくれる(はずなだけでなあ…)

・水晶発振子

VS1011e が動くにはクロックが無いといけない。だからクロックを作ってくれるこれが無いとマイコンがいくら頑張っても何も起きないという実はマイコンより偉いかも知れない部品

・レギュレータ

上にあげた三つの部品はどれも動くには電気を流さなければいけません、ただ闇雲に流せばいいというわけではなく決まった電圧の電流を流さなければいけません。しかし、コンセントの交流を直流に直すために物無で使っている電源はほぼ全てが 5V の出力。これを上の部品たちに流すと break します。そこで登場するのがレギュレータ、電圧を変えてくれる部品です。要するに実はマイコンより(ry

・他もろもろ

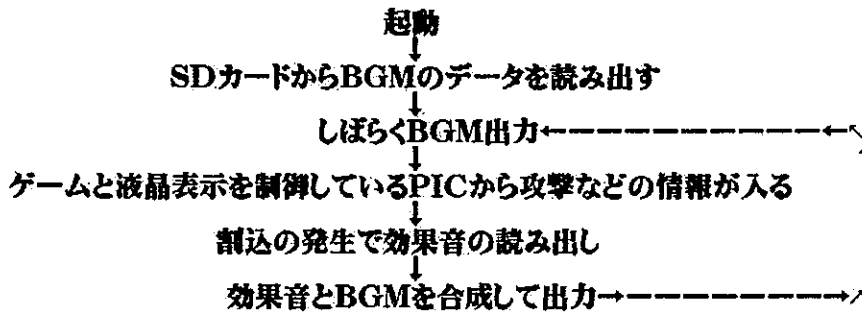
DC ジャックやイヤホンジャックといった接続用の部品と抵抗とかコンデンサといったほとんどの基板についている部品。LED は製作中につけましたがもし光っていても直接の関係はないです

5、おもな動作

SPI 通信による SD カード読み込みと MP3 デコーダの制御、

SPI 通信は基板内での高速シリアル通信を可能にする通信方式ですが、ノイズに弱い実体は基板内でも離れているとうまくいかなかったりします。基本的には Master が常に主導権を握ったうえで Master 側と Slave 側に分かれて三本の線(SCK・SDI・SDO)、また必要に応じて SS の線を一本から複数本使用して Master から Slave へ出力しそれに返信する形で Slave から Master へ出力するという循環したデータのやり取りを行っています。データの送信は Master が必ず先行するので Slave からデータを送るにはダミー信号を送る必要がありますがさっぱり分からないのでこれは割愛させていただきます。

6、フローチャート

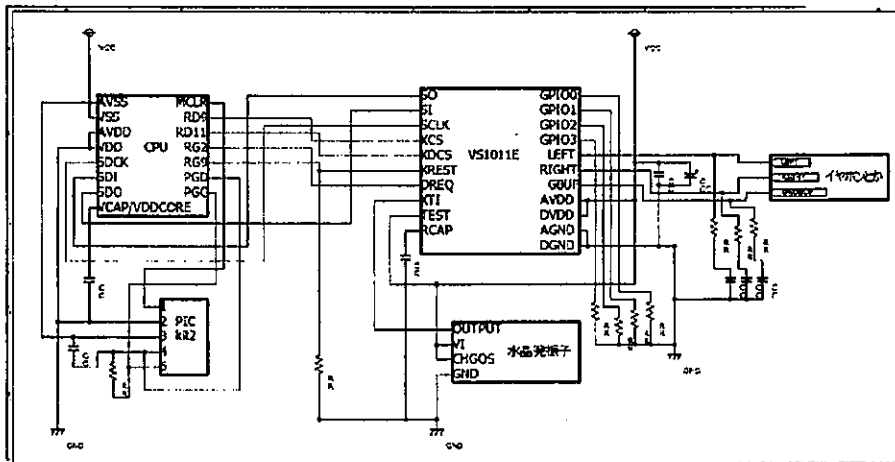


7、感想

今回の僕の製作ではMP3の再生を目的にはじめてプログラムに挑戦しましたが、基板やプログラムなど構成要素が増えるのとどこに問題があるのかが分かりにくくて、僕のほかにも何人かの先輩がやっていますが、僕は現時点の三月二十日でも未だに一定周波数の音すら出せないでいるのでBGM位は出せるようになってほしいです。

来年プログラムをやることがあれば基礎部分はさっさと終わらせたいです。

8、回路図



※この回路図では使用していないものをはじめとしたいいくつかのピンと未完成なSDカード部分を省略しています。

※なお、ここに掲載されているフローチャート・外観図・回路図は変更される可能性があり、またこのまま使っても正しく動作するとは限りません。

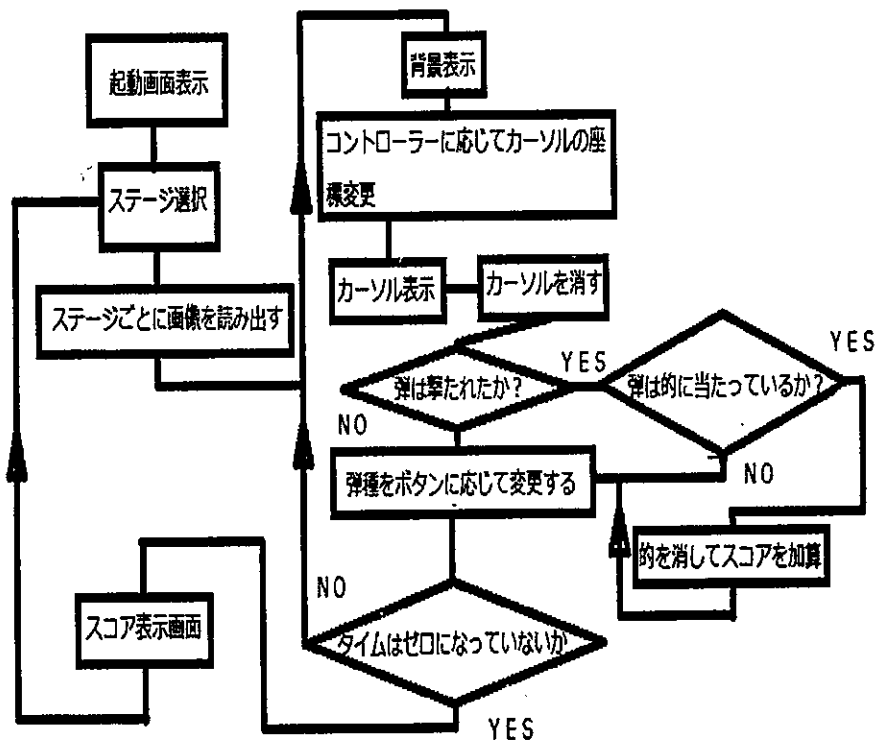
・ 液晶表示部分

兼歴史捏造担当:H1山本

9、プログラム

今回マイコンに 32MX440F512H、液晶は2.4インチの QVGA カラーグラフィック液晶を使っています。発振は内部発振でやっています。

申し訳ありませんが、マイコンと液晶のピン配置はプログラム次第で変えることが可能なので割愛させていただきます。プログラムについてですが、全部書くには長すぎるのでメインのフローチャートだけ載せます。



10、やっていること

一応今回は「C言語で液晶表示をしてあわよくばSDカードとの通信をし、さらにできればテレビ接続でもしようかな」とか思っていましたが、自分の知識と実力では到底及ばず色々教わりながらなんとかマイクロSDカードから画像データの読み出しをして、液晶に表示するところまでできました。

マイクロSDカードとの通信にはSPI通信を使っています。SPI通信とは上のほうで四柳君が説明していますが簡単に言えば少数接続端子で通信を行う方法です。

またマイコン用のファイルシステムはフリーソフトとして公開されている「FATfs」というものを使わせていただいております。

四柳君の音響部分と液晶表示部分の接続はこれからですが、BGM4曲効果音4種を目指していますので、4本の配線での通信を予定しています。

11、アナログジョイスティック

このゲームのカーソル操作にはアナログジョイスティックを使っています。

アナログジョイスティックとはPSPや3DSについているやつです。まあ、ボタンとは違い感覚的に操作する感じのコントローラーですね。

アナログジョイスティックの仕組みとしては中に縦軸、横軸の二つの可変抵抗が入っていて、この抵抗値の変化によって動きを読み取るものとなっています。と、まあぶっちゃけこれだけなんですけど、一応もう少し。アナログジョイスティックの二本の出力は電圧の変化でしかないアナログデータなので、A/D変換をする必要があります。A/D変換とはその名の通りアナログデータをデジタルデータに変換することです。A/D変換という機能自体は周辺モジュールとして組み込まれています。

こうしてアナログデータを数値化することでカーソルを動かすことが可能となります。

12、感想

今年初めてC言語を使い、液晶を使い、SDカードを使いと新しいことをやり始めたのはいいものの、わからな過ぎてどうしようもありませんでした。そんな時色々というかほぼすべてを教えてくださいました物無の皆様。本当に感謝しています。

まあ、上記の通りあまりよくわからずにやっていたため自分に知識があまり身につけていない気がします。最初に作るものを決めたときに明確に作りたいというものがなかったのもよくなかったですね。

今回、製作のレベル的に周りに追い付きたかったけれど、全部において頼っただけになってしまいました。来年はもうちょっとほかの人と違うことがしたいですね。しっかり勉強をしてやりたいことが出来るように製作に臨みたいと思います。あれなんか去年同じこと言っていたような・・・。

花火師

製作者 M2 石倉 匠

設計者 H1 岸田 聖生 さん

協力者: 物無の皆様

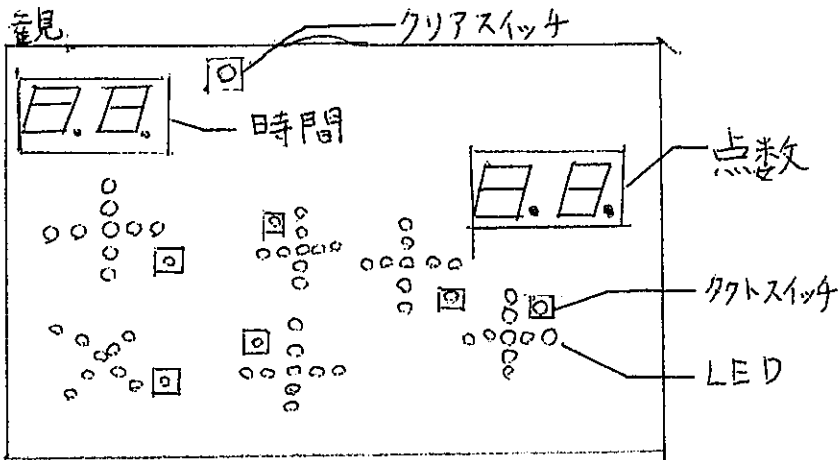
ロロン電筒う。

ルール

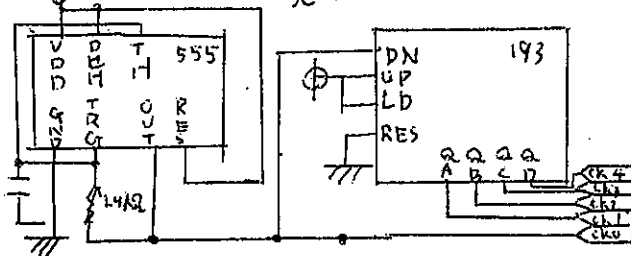
LEDが内側から外側へと光っていくので、

一番外側に光が届いた時にボタンを押すとポイントが付き、
それ以外の時にボタンを押すと、ミスになってしまいます。
制限時間に何ポイントたまるか競うゲームです。

外観

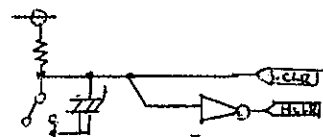


回路図

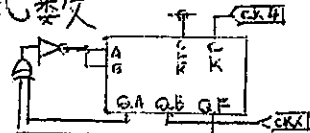


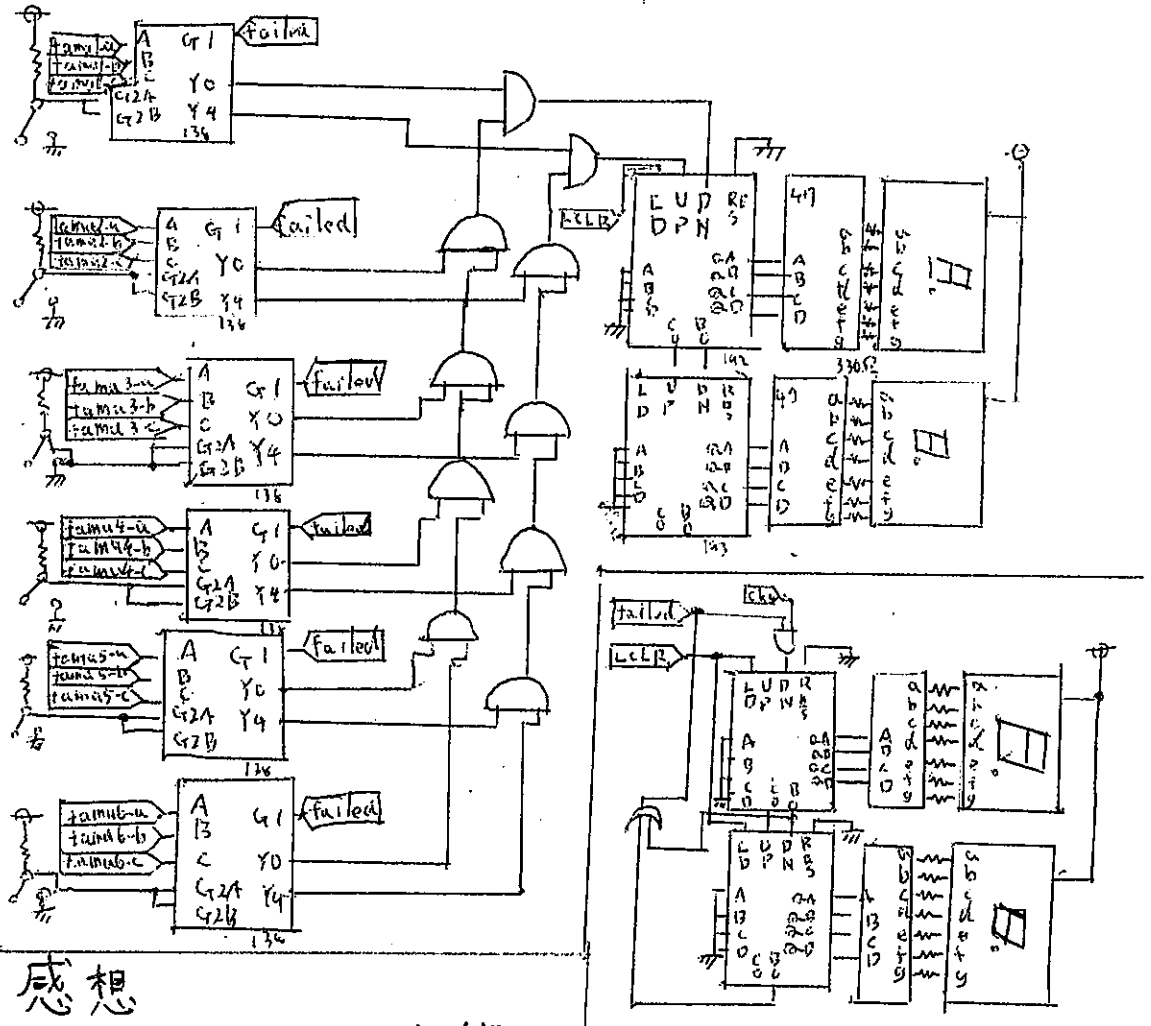
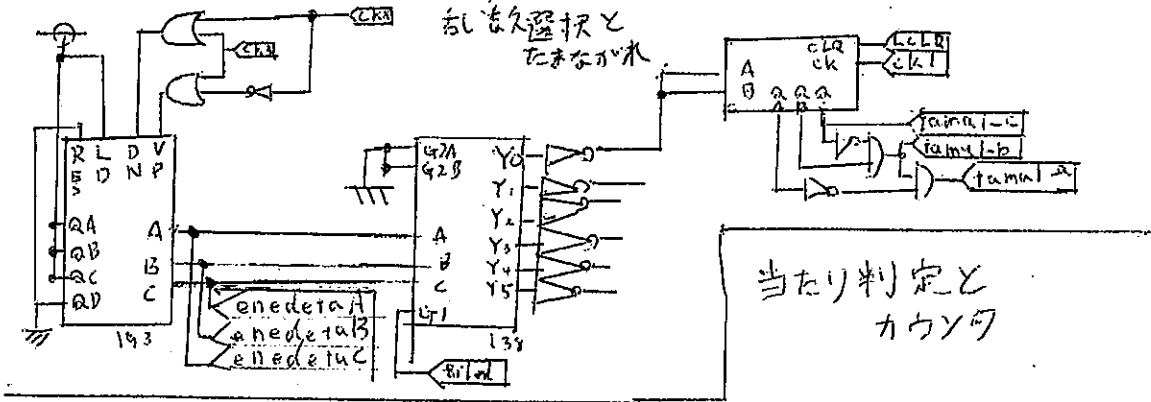
発振

リセット



表示数



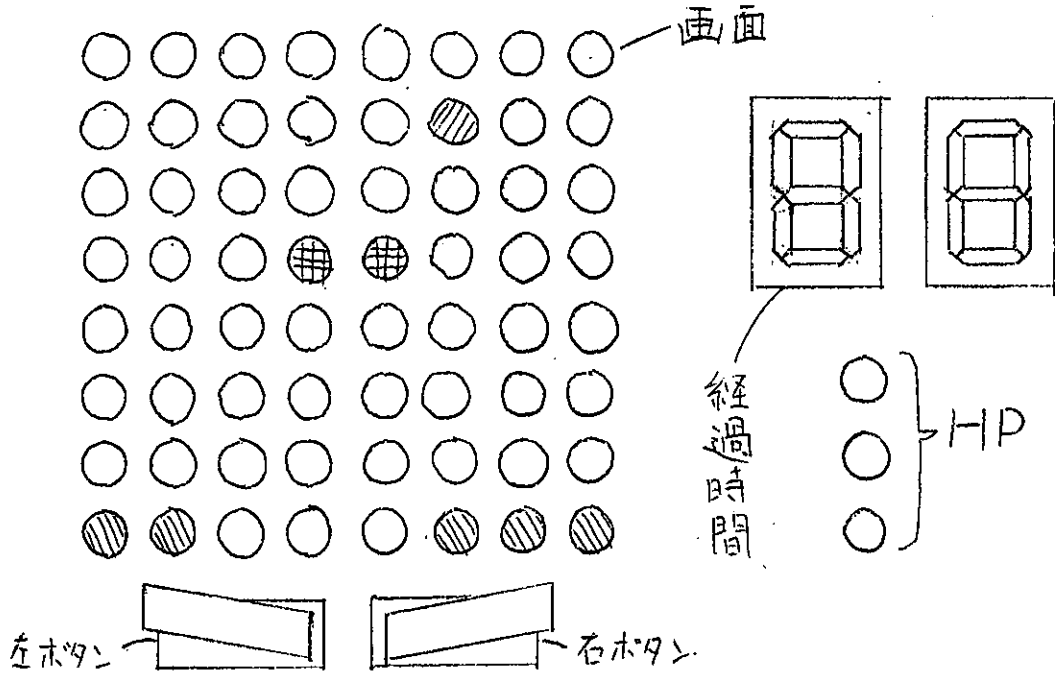


感想

僕はハンダづけが苦手で、金同はんだがはがれて
しまうことが多くありました。配線が楽だった
ことや、岸田さんの助言などにより無事完動する
ことができました。

氷を守る船台

外観



ルール

上から障害物(1マス)とかや(2マス)が落ちてきます。
また、下1列は自機(5マス)が左から右にルーラーしています。

自機に、障害物とかやが当たると、HPが減っていきます。HPが無くなるまでゲームオーバーです。あなたは左右のボタンでかやを左右に動かす事ができるので、かやより早く落ちてくる障害物を受け止めながら、自機に当たらないよう落としましょう。

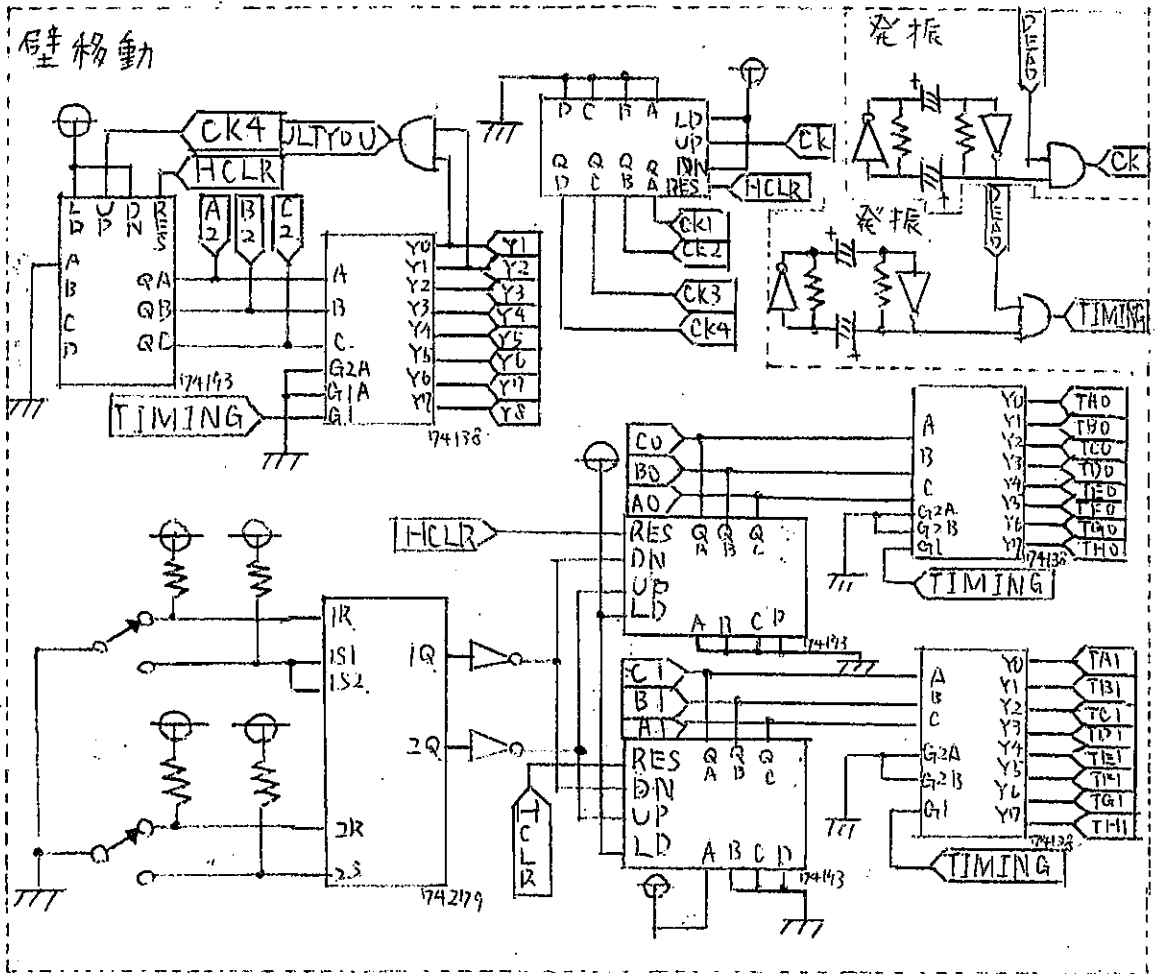
HPは3つです。ゲームオーバーになるまでの経過時間があなたのスコアです。
時間ごとに流れてくる速さが速くなります。

ストーリー

設定は、宇宙から味方の戦闘機が、降ってくるミサイルを迎撃時から母船に帰ってくるという壮大な物語です。

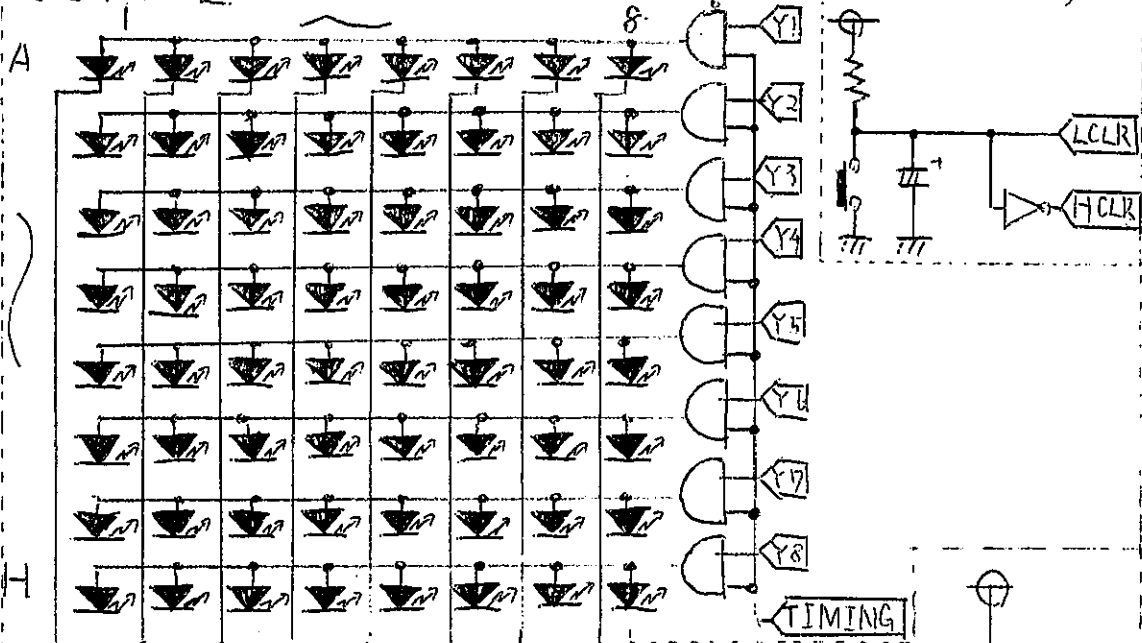
あなたは宇宙へと飛び、攻撃されている母船を守りながら帰還して、英雄となるのです！
 時間とともに強くなる敵にも屈せず、母船を守り抜きましょう。
 ※本ゲームでは飛び立つ部分はありません。

回路図

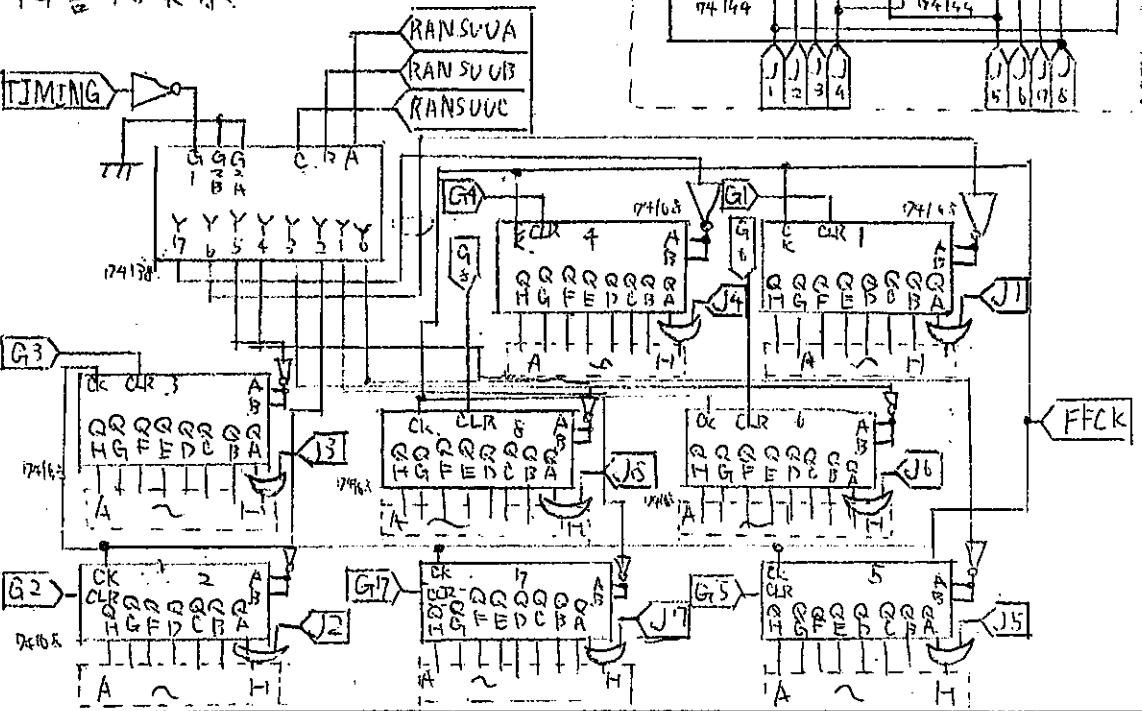


LED 配置

71) 32144

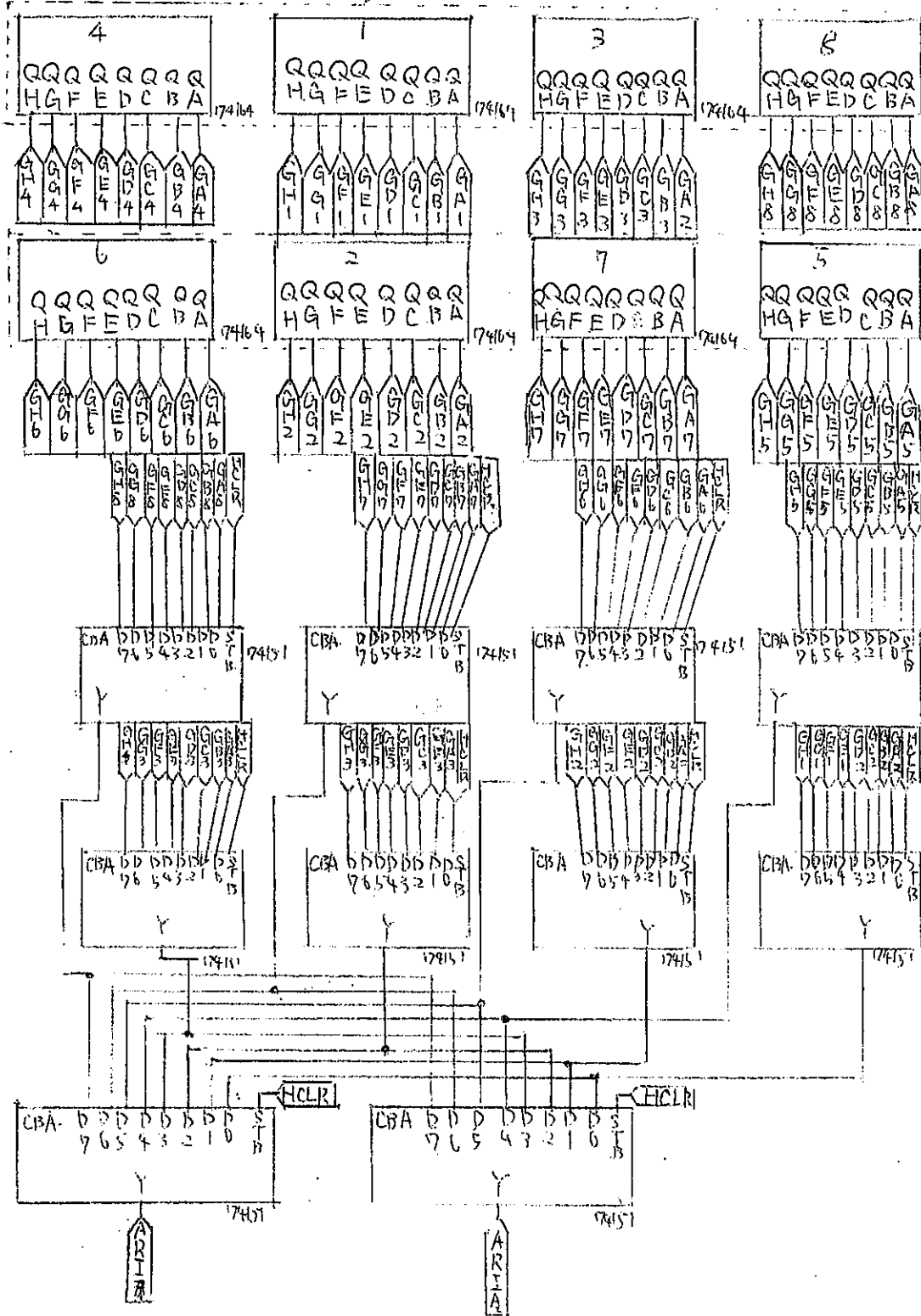


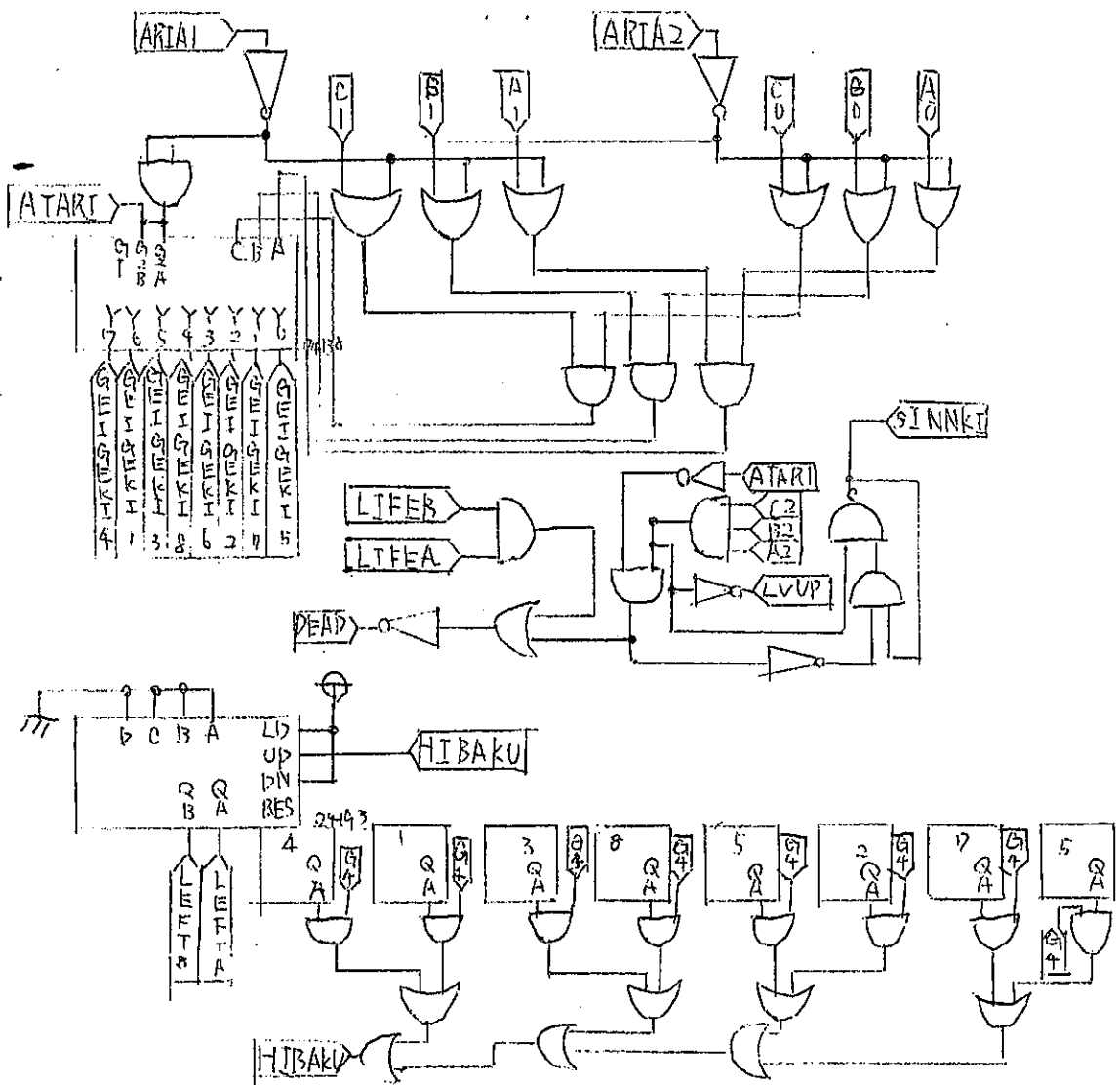
障害物表示



あたり判定

障害物と同じ





感想

製作は、山本さんがわかりやすくおしえてくれたので、内容を理解することができ
うれしかったです。

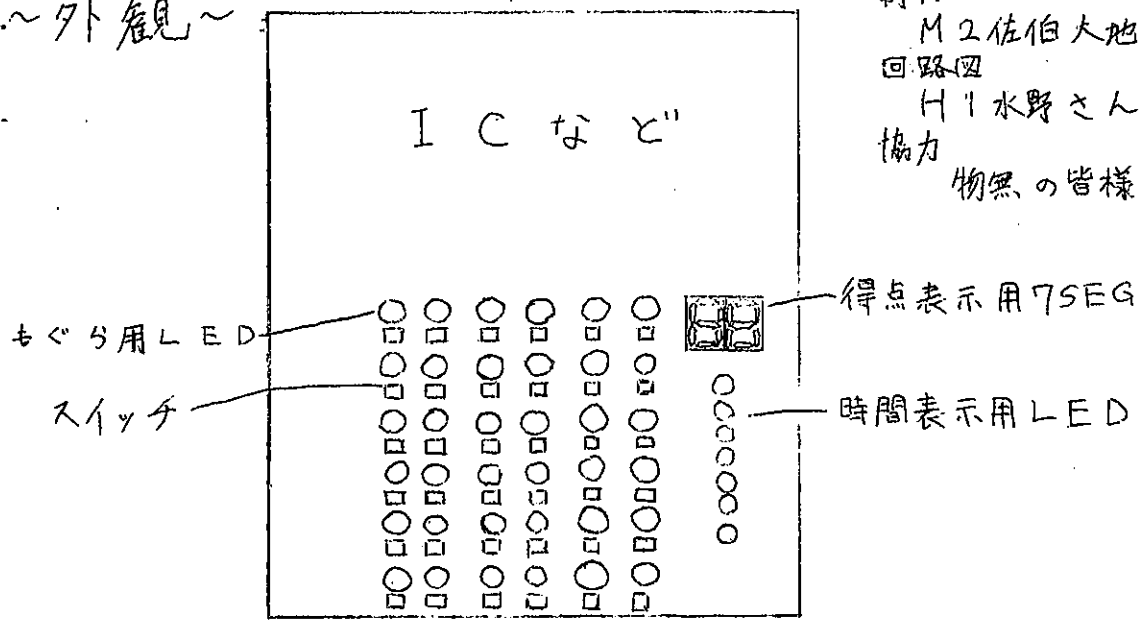
しかし、半田づけの技術は向上せず、1ヶ所が発生したり、ショートすることがしばしば
でした。この点は来年には改善したいです。

1年間の活動はどれも充実したものでした。喋って1年間活動できて、良かったです
思います。

土電叩

制作
M2佐伯大地
回路図
H1水野さん
協力
物無の皆様

～外觀～



～ルール～

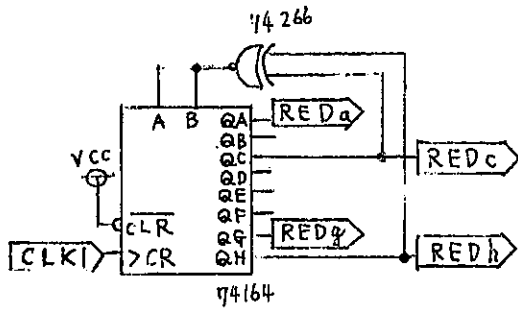
もぐら用LEDが赤や緑に光ります。赤が光ったときにそのLEDに反対した(LEDのすぐ下の)スイッチを押すゲームです。赤を押すと得点が+1、緑を押すと-1となり、赤と緑どちらも光っているLEDや何も光っていないLEDのスイッチを押しても得点は変わりません。時間表示用LEDが全て消えたかゲーム終了です。

～ストーリー～

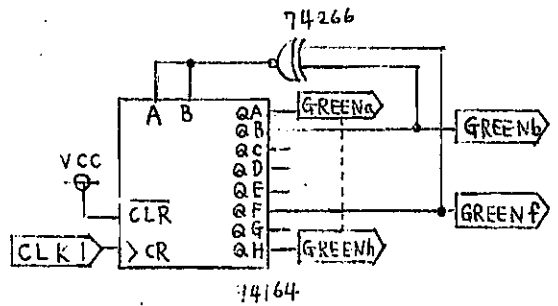
ある晴れた日の朝、家でテレビを見ていたさとしは、226番道路に赤い色違いもぐらが大量発生したというニュースを見た。しかし偽の緑のもぐらも大量発生しているという。さとしは幻の赤もぐらをつかまえるべく、立ち上がったのであった。

～回路～

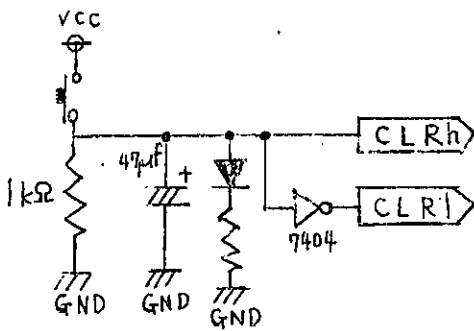
乱数回路赤



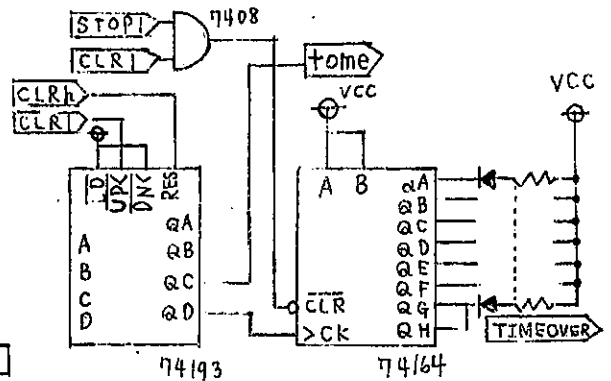
乱数回路緑



クリアボタン

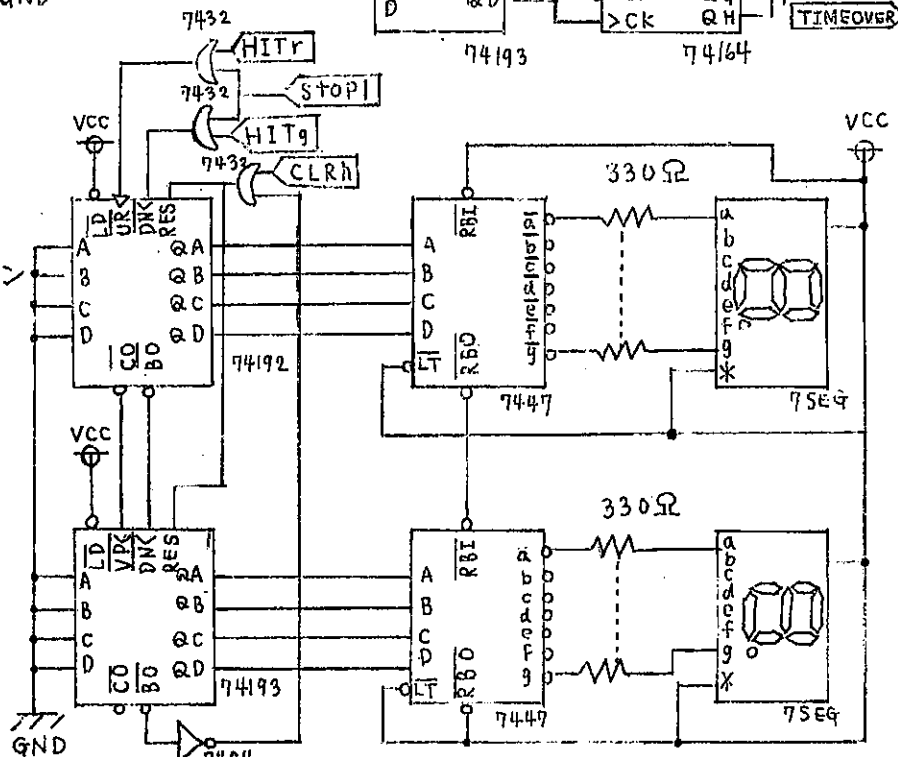


時間表示

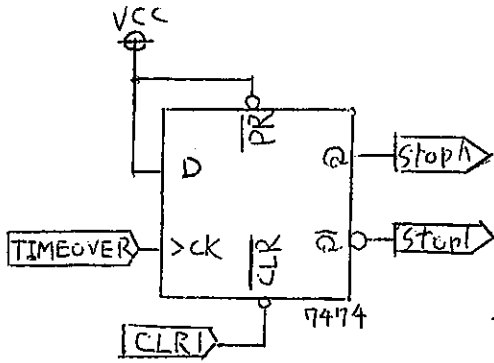


点数表示

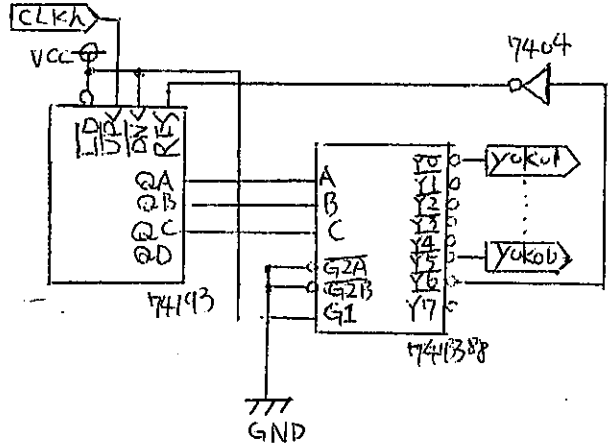
7セグは
アノードコモン



制限時間終了

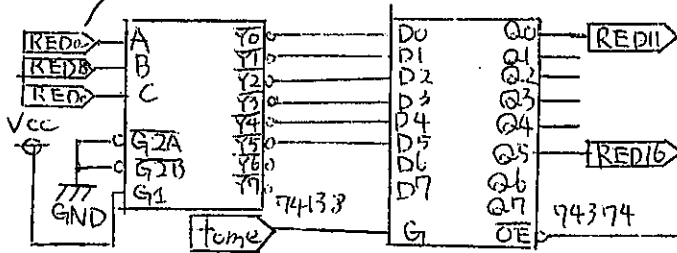


74インテリク点灯



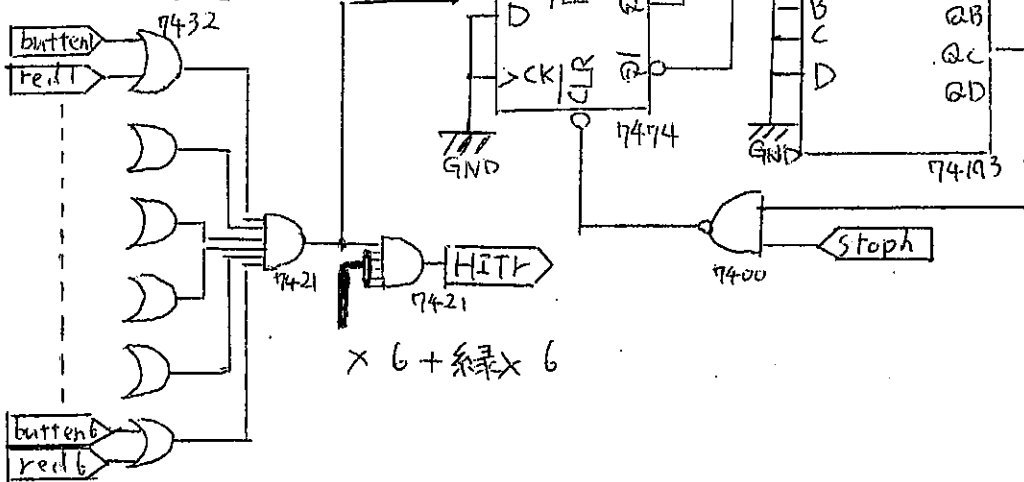
敵表示赤

abc, def, gha, bcd, efg, hab



$\times 6 + \text{系録} \times 6$

当たり判定赤



$\times 6 + \text{系録} \times 6$

感想

とにかく当たり判定を動かすのに時間かかりました。全体の半分くらいの時間を費してしまいました。でも早めに完動できたのでよかったです。来年もこれくらい余裕を持ちたいです。

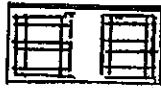
Space Aegis

○ 外観

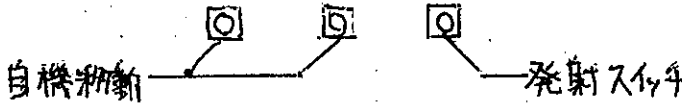
残り時間



点数



(1) ○ ○ ○ ○ (2) 自機



製作M2: 藤川
回路図H1: 岸田さん
協力: 物無の皆様

○ ルール

敵(赤)と味方(緑)が止から流れてきます。IIの範囲で自機を移動して敵をよけ味方を回収します。発射スイッチを押すと、自機から上に向かって玉(緑)が発射されます。玉は敵か味方に当たると、爆発して縦一列の敵と味方を消します。自機は1→2、2→1の移動が可能です。敵をよけたり撃ったりしながら味方を回収してください。

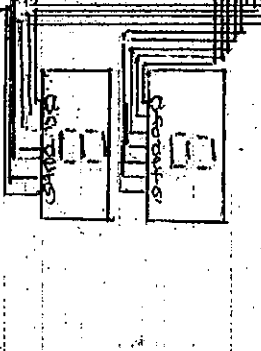
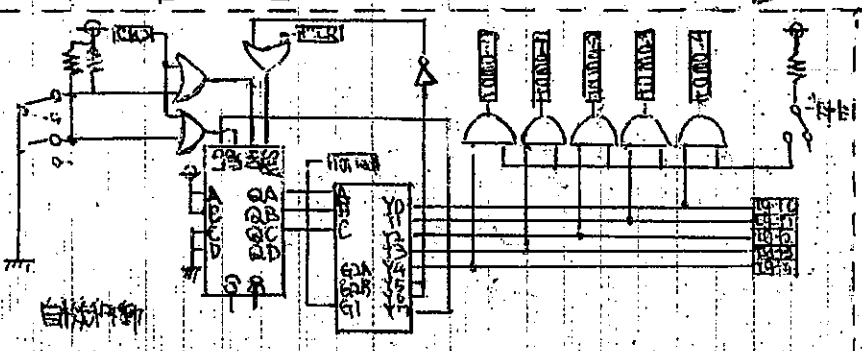
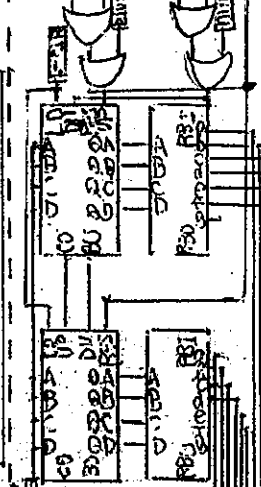
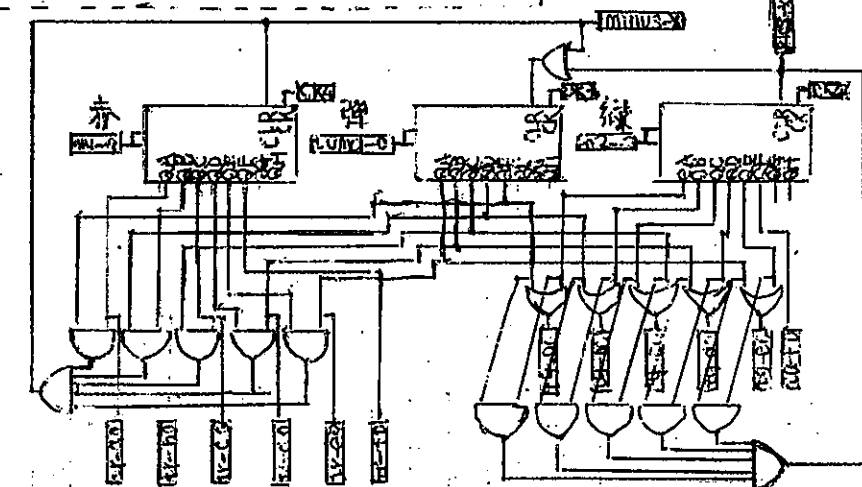
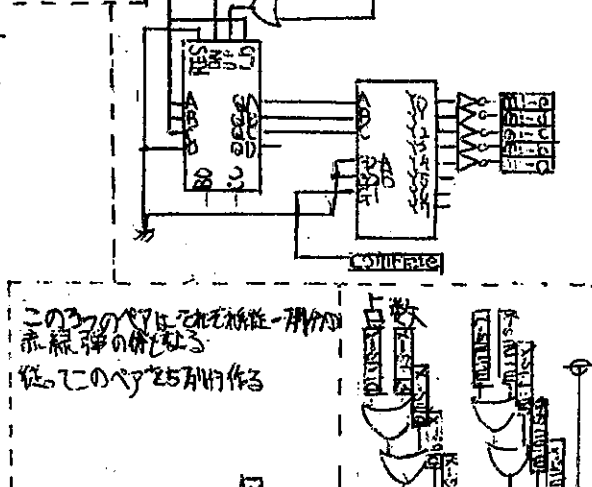
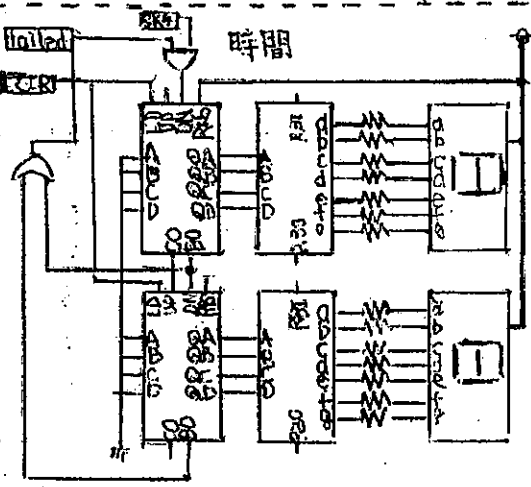
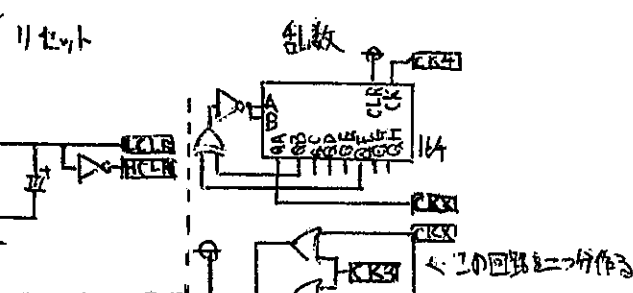
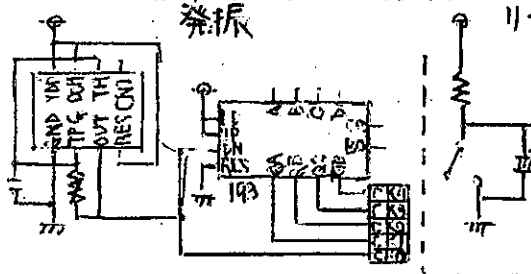
○ ストーリー

時は25世紀。人類は太陽系全体に広がろうとしていた。しかし謎の生命体が現れ地球の侵略を開始し、人類はまもなく間に全滅寸前まで追い詰められた。だが人類には切り札が残っていた。直線上にある物を全て破壊する大砲をうみ果なる二点を瞬時に移動できる「Tエンジン」搭載した宇宙船である。この切り札いで味方を救出しながら最後の人類は強大な敵と立ち向かっていた。

感想

配線のミスが多くまだ完動できていないので来年からミスのなくできるようになります。

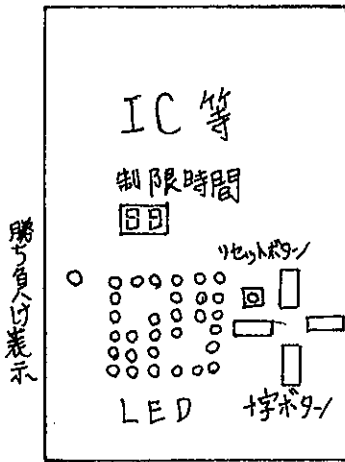
回路図 以下の通り作って動作を確認し完成
 発振



SNAKE

制作者 M2 宮嶋 優大
回路図設計者 H11 釘 真太郎さん
協力 物無の皆様

1. 外観



2. ルール

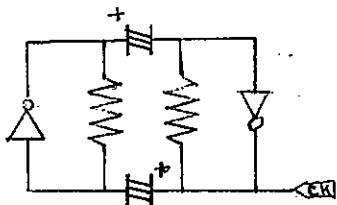
十字ボタンをおして自機を操作し、へじからにげるゲームです。かべにあたらないようにランダムに動くへじからにげます。そして制限時間内に宝を5つとります。かべかへじにあたるか、制限時間内に宝を5つとれなかったらゲームオーバーです。5つの宝をすべてとったら、ゲームクリアとなります。左上のLEDが赤くひかたら負け、緑色にひかたら勝ちとなります。

3. ストーリー

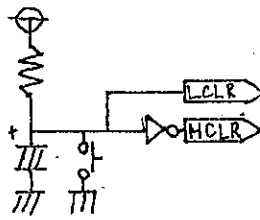
考古学者であるジョン博士はすべてを黄金にしてしまう伝説の財宝を探索するために発掘をすすめていた。するとつぜん大きな穴があき、みな落ちしまった。運よく生き残ったジョン博士は穴の先に続いていた細い道を奥へと進んでいった。するとそこには広し空間があった。つぜん大きなアゴコンダがおそって来た。このままだとへじに食べられてしまう。ジョン博士は死を覚悟で宝をひろいにいった。

4. 感想

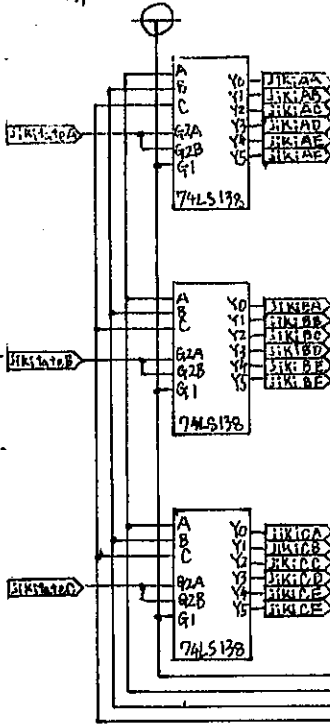
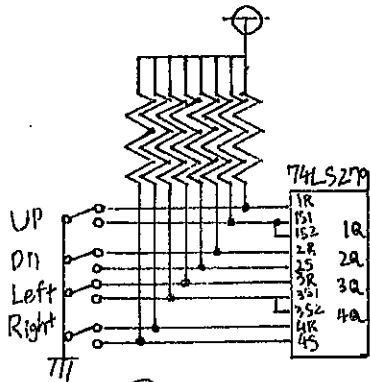
完成するのがおそくなってしまったのは、ICの数が多いためというところもあるが、配線やハンダ付けをするスピードがおそかったからだと思います。出席日数も多い方ではなかったのをご反省しています。これからもがんばっていきます。



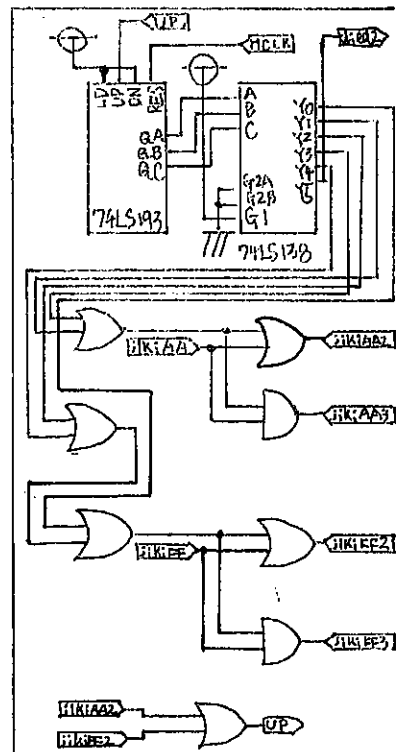
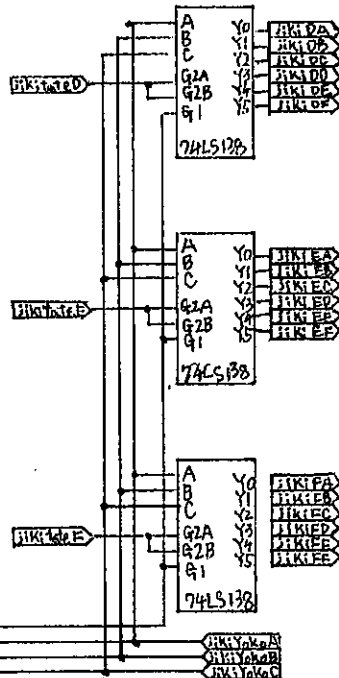
進



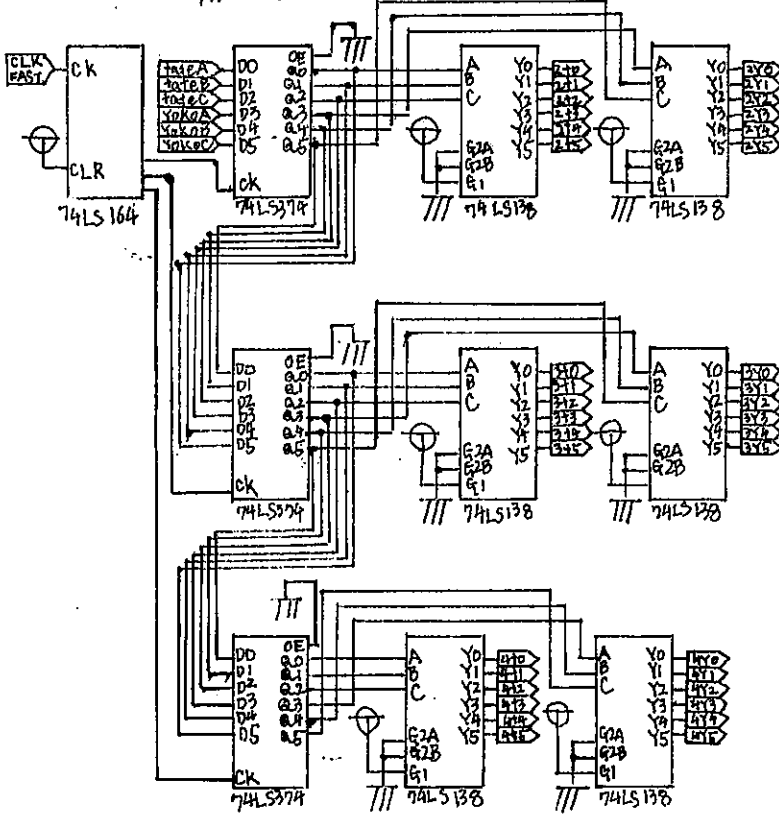
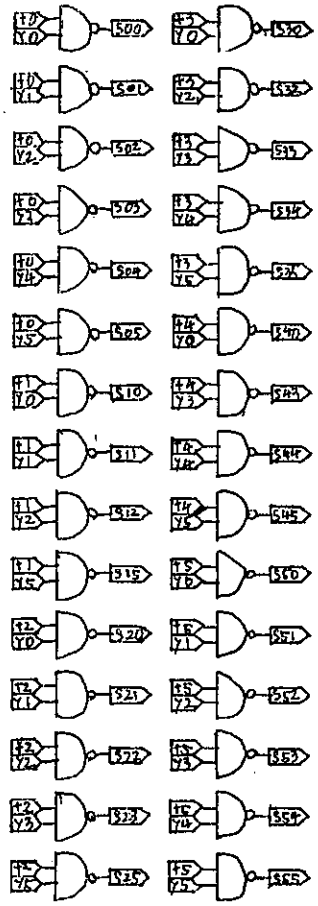
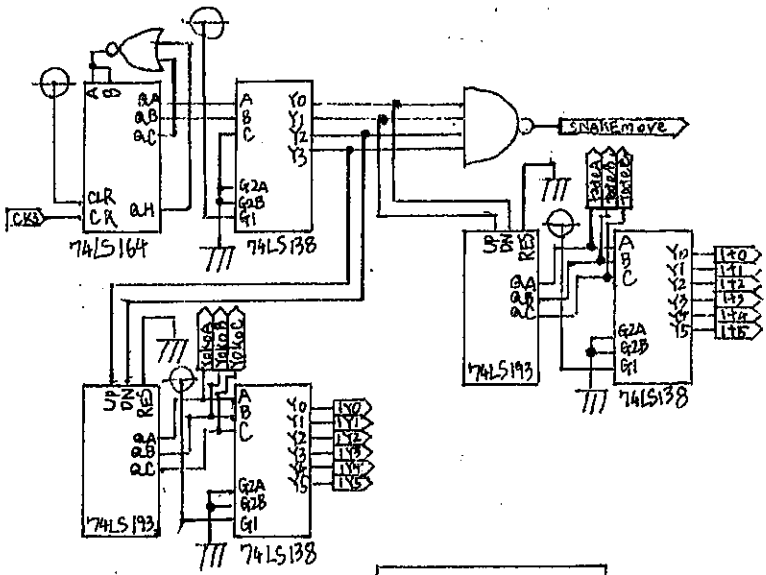
713



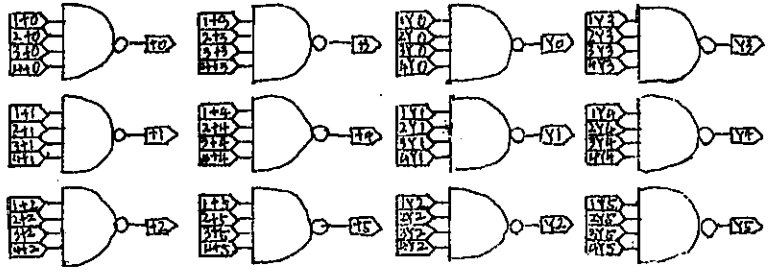
自動移動

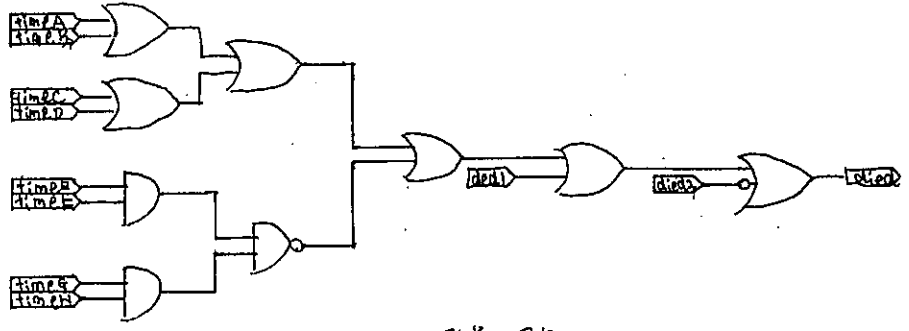
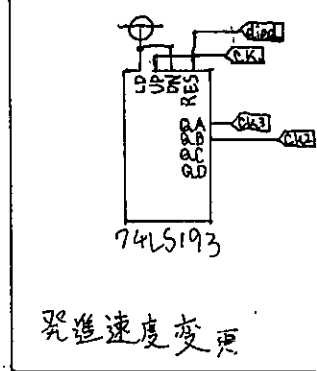
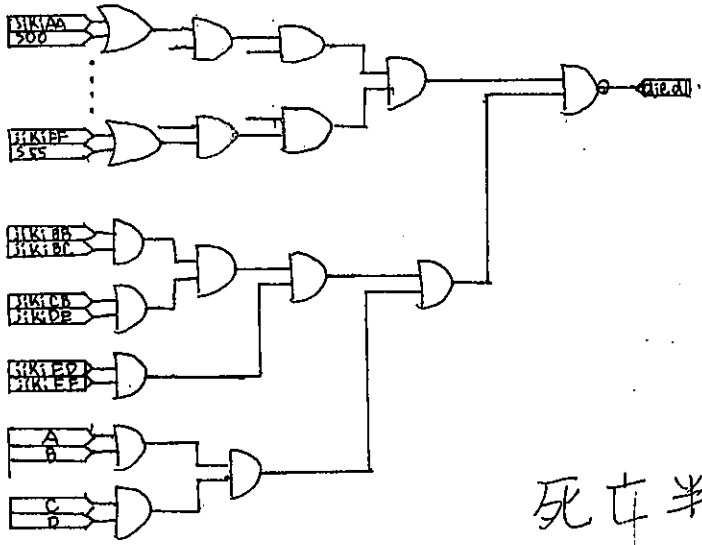
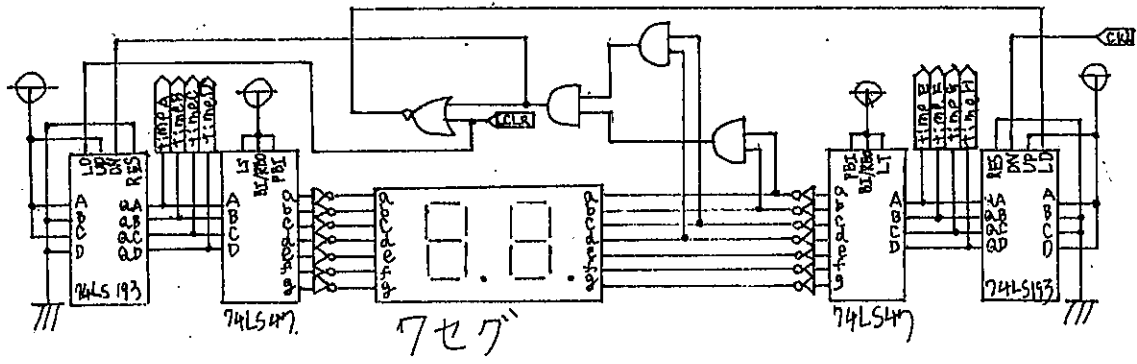


宝表示



へじ移動

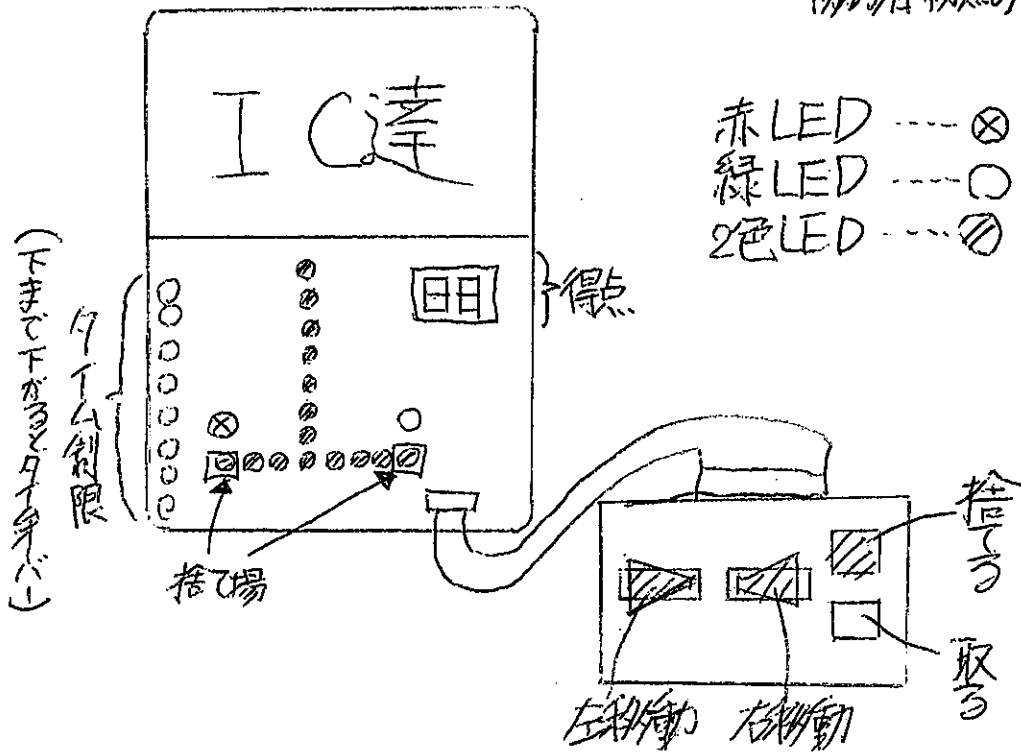




塵袋と科学大衝突

1 外観

製作者 1/2 横尾幸丸
 回路図設計者 H1 森一以
 協力者 物無の啓康



2 ゲーム内容

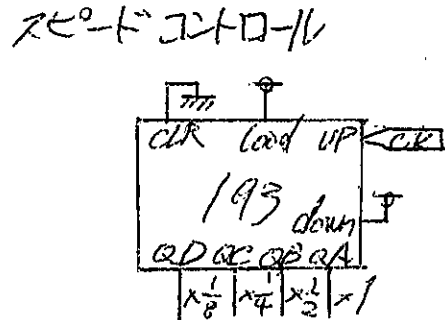
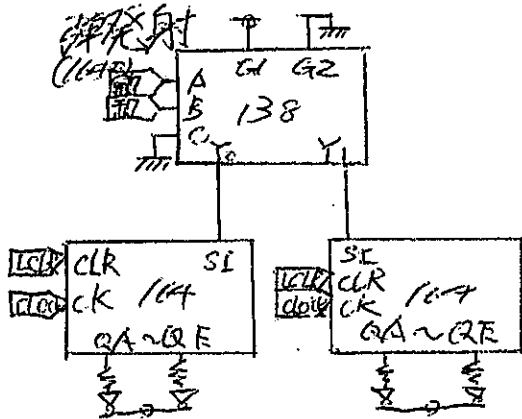
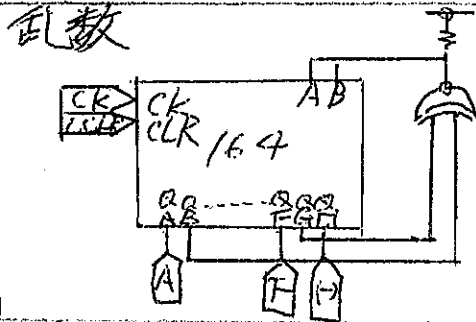
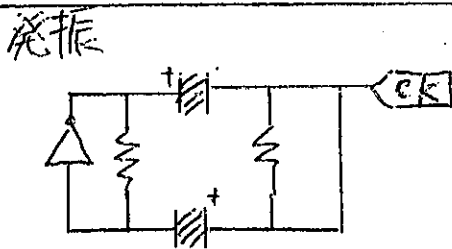
このゲームは「おおざらば」に似た、上から降ってくる玉を自機が取り、左右にある捨て場に捨てるゲームです。
 まず、上から赤と緑の玉が降ってきます。それを左右に移動し、玉と交叉点に移動させ、取るボタンで取り捨てる。そして緑は緑の捨て場へ。

赤は赤の捨て場、移動させ、捨てるボタンで捨てます。ほとんどのポイントです。
 時間制限は左側にあるLED表示。一番下に赤がいくとゲームオーバーです。

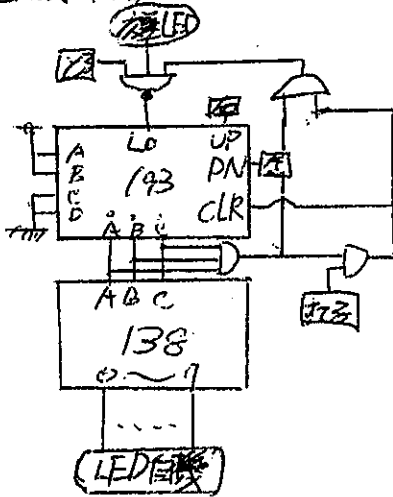
3 ストーリー

とあるモンスターがある日、腹をかせ、ある都市へやってきました。
 その都市にはたくさん人のいる食べ物があり、また木を
 垂らしたから歩いていると、
 「^{クワッ}者を見 ^{クワッ}者を見!!」
 という声が聞こえたかと思うと、複数人の腕章をつけた人達が
 「いまず、この場所から立ち去りなさい」
 と言われた。
 もちろん、腹がすいているため、無言で歩き続けると、炎や氷や電撃
 などがその人々から飛ばされ、命に危険をきたした。
 ひたすら逃げ続けたモンスターは考えた。どうすればあの攻撃に
 対処できるか？ すると頭の上にリングが落ちてきた。
 上を向くとリングの木があった。そこでモンスターは思いついた。
 この木を使って、スバヤサを高めよう……。

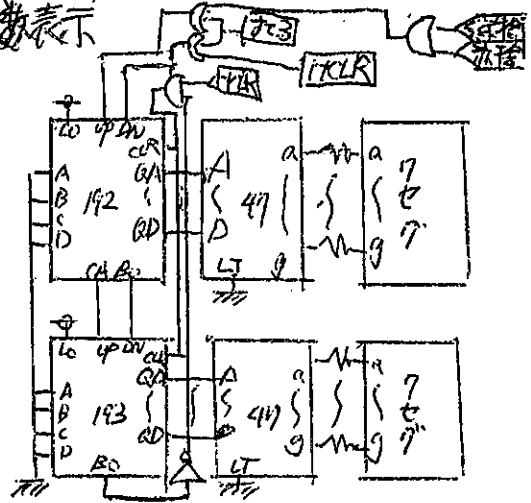
4 回路



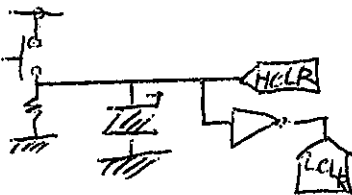
自機自動



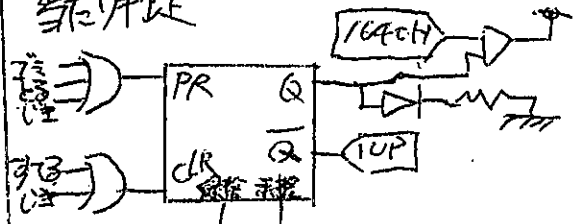
点表示



クリアスイッチ



当たり判定



5 感想

正直本当に作るのは難しかったです。ハンダづけもいづもいもハンダで大変でした。

まだまだ物無員としては未熟者ですが、物無の皆様、僕の両親、在校生、そして本を読んでいる皆様、これからもお願いします。

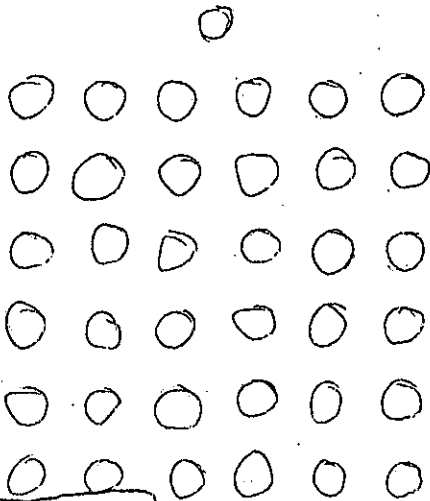
きよなる

製作者 M2 山本 淳太郎
回路図設計者 さん
HI 金子 さん
協力 物無の さん

百の目

概要

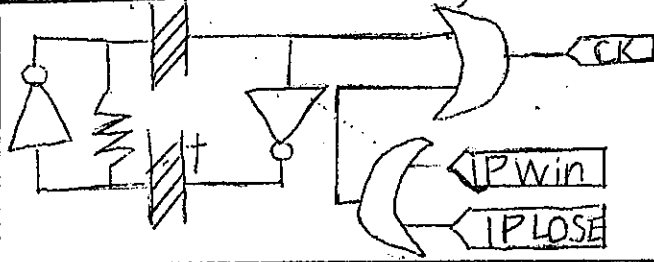
目目



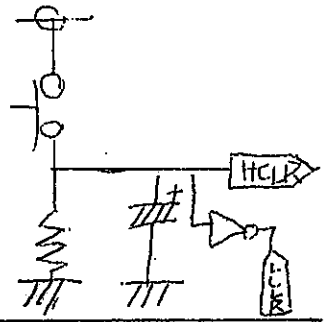
ストーリー

今日は8月31日僕はかんばん
で宿題をやろうとした。だが、
なかなか宿題が終わらない。お
そいかかってくる宿題を君はか
おせるか。

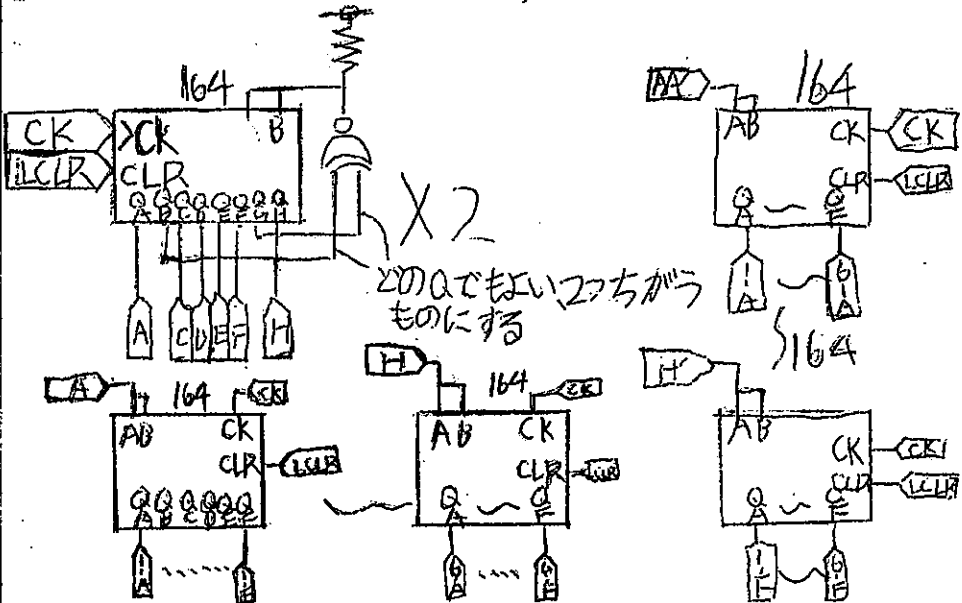
回路図 発進回路



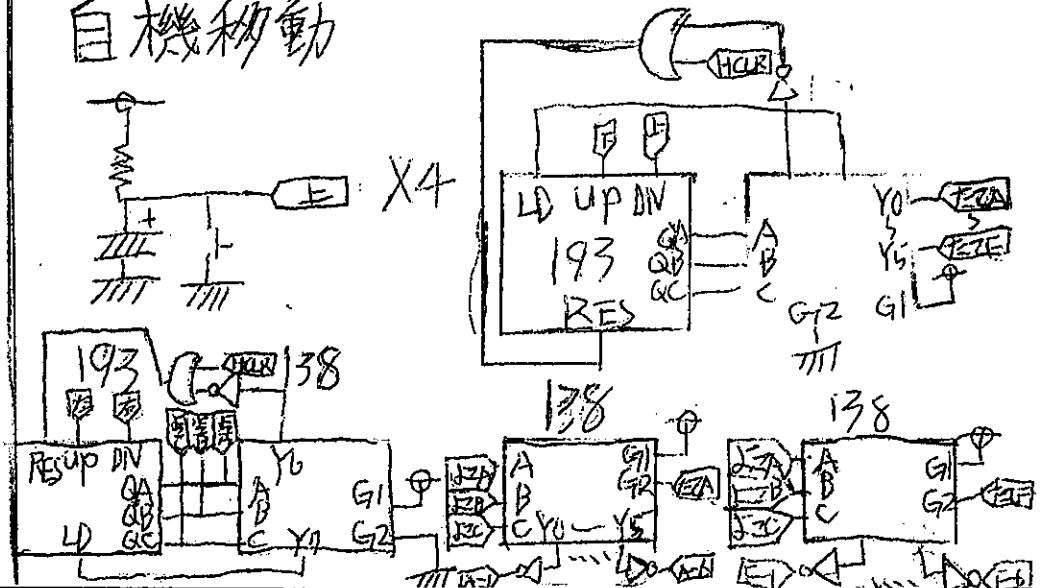
クリアスイッチ

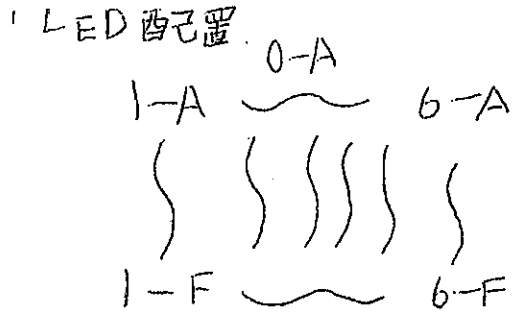
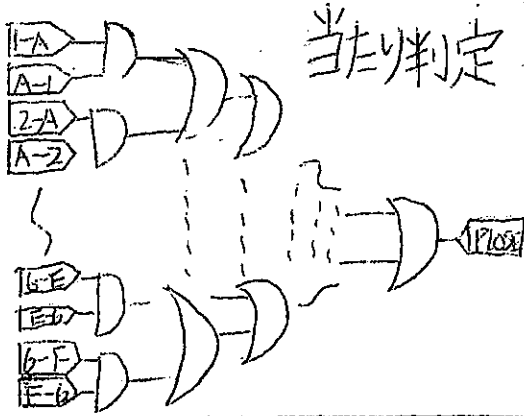


乱数回路とシフト回路



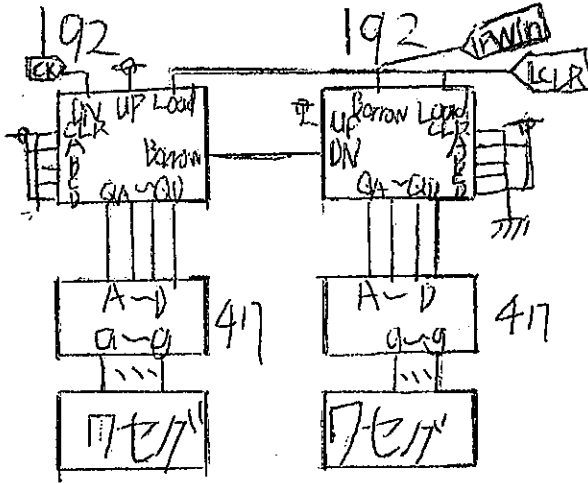
自機移動



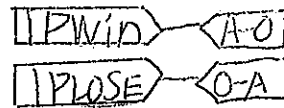


二色LEDの色は数字英語
どちらが先かで区別する。

時間



勝敗



ルール

流れてくる球をかき
します 99秒経過
れば勝ちです。当たる
と負けです。

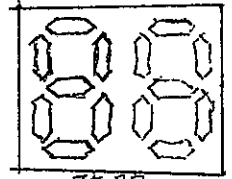
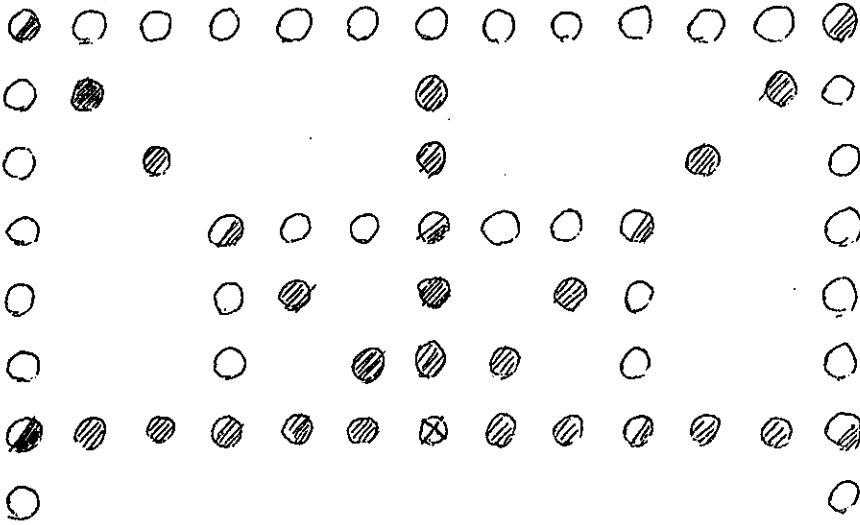
感想

製作では配線を何十本も外したり、ICをとったり、銅板をはかしたり、いろいろと大変なことがありました。それを今度の製作に生かし、来年も楽しく製作したいです。あと今年は無面目にやらかさず、話ばかりしていたので少しづつ真面目にしていきたいです。

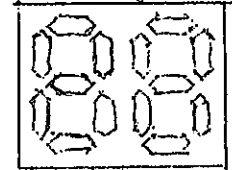
北 水 陣

製作 M2 渡辺 悠木
 回路設計者 H1 河村 さん
 協力 物無 の 皆 さん

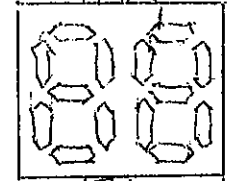
1. 外観



時間



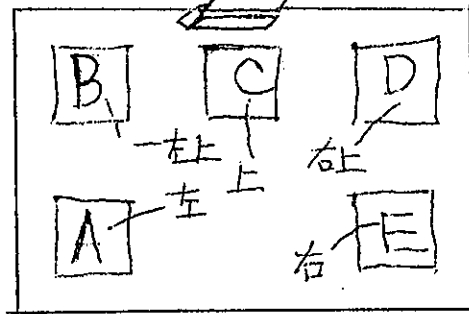
弾数

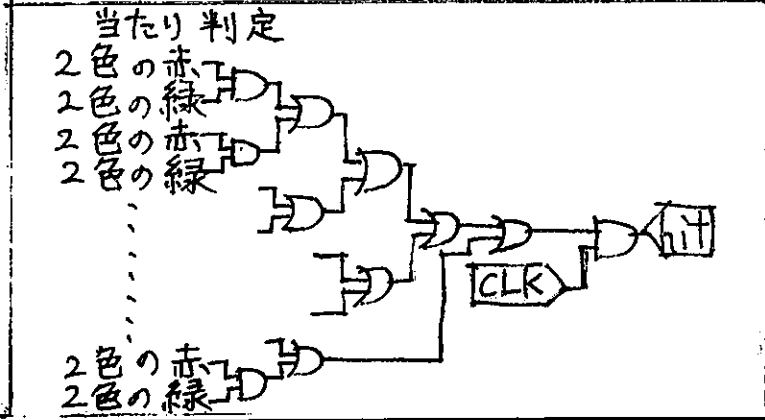
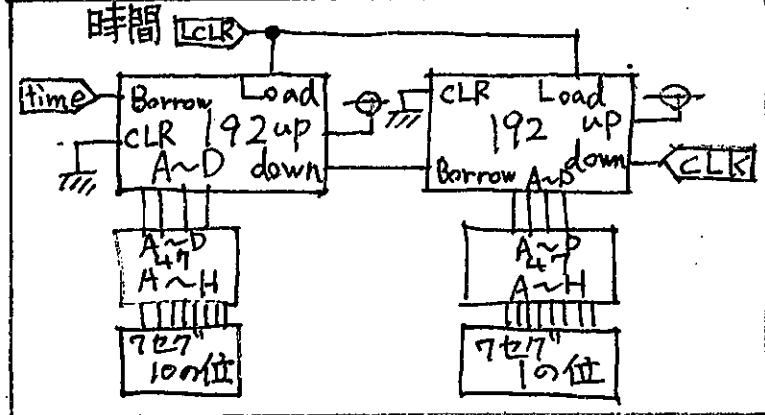
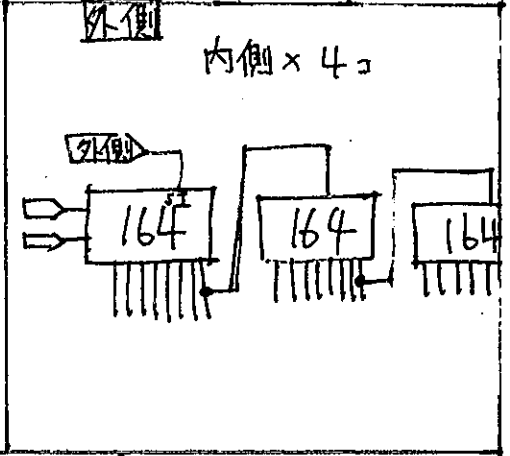
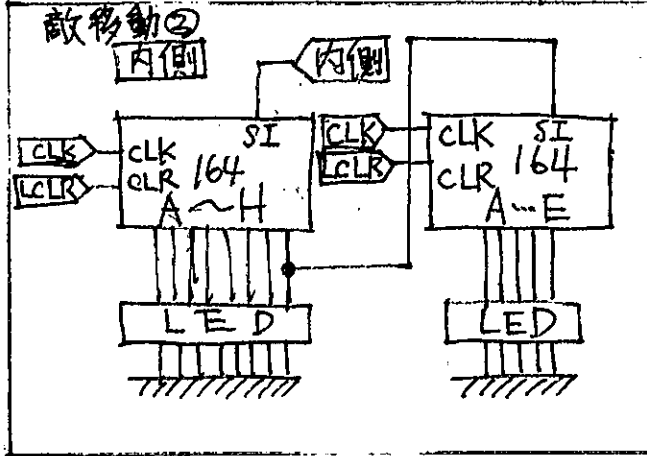
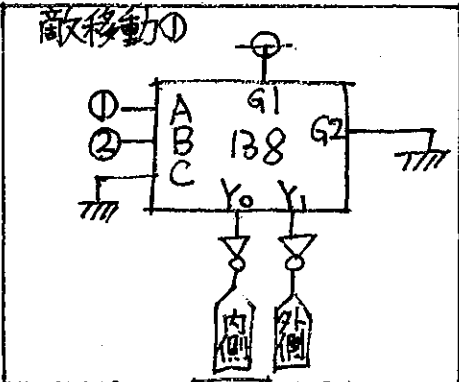
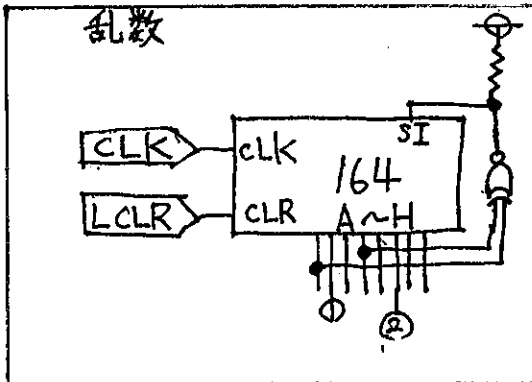


得点

○...敵(赤) ●...弾(緑) ◐...敵かつ弾(2色) ⊗...自機(青)

コントローラー





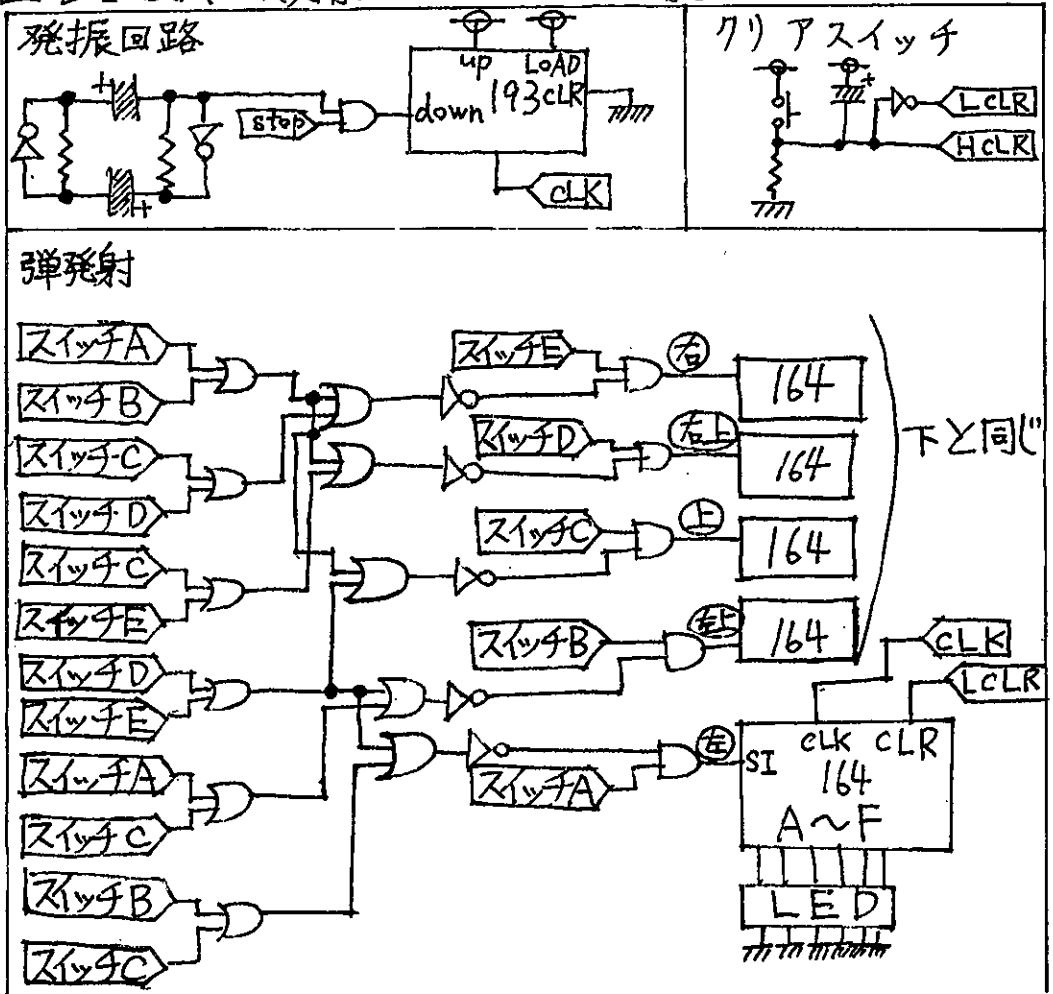
2. ルール

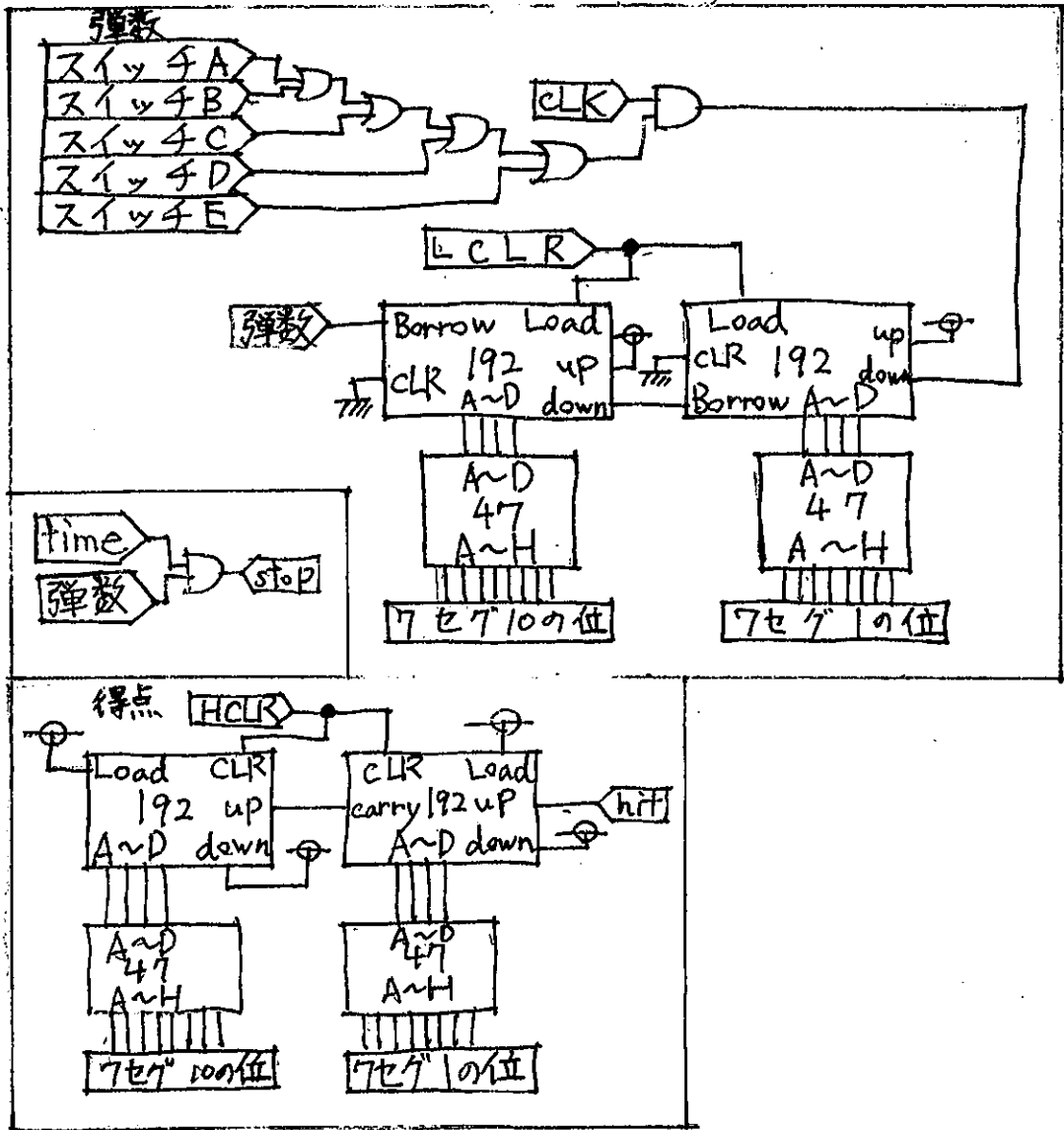
自機(青LED)から五方向に向かって、弾を発射します。右から左へグルッと敵が移動しているので、そいつらに弾を当ててやりましょう。当たると1点もらえます。弾は30発しかないのでねらってつちきましょう。長押しは弾が減り続けます。同時押しは弾が出ないので注意。

3. ストーリー

彼は逃げ出した。そこはゾンビ生成工場だ。ゾンビになりきれなかった彼は30発の弾と銃を持って逃げることにできなかった。暫くして彼はゾンビに追い詰められてしまった。後ろは海であり、逃げ道はない。仕方ない、この銃を使ってまた逃げ出すしかないな....」

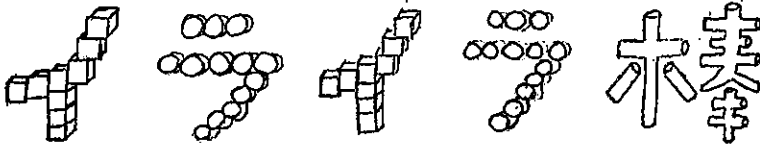
4. 回路各図(一部実際の回路とは異なる場合があります。)





5. 感想

入った頃は、「はんだごて」という単語も知らなかつた。僕ですが
 既に一年程経ち、物無にも慣染めてきました。製作は、
 僕のはとても簡単で一番完動させることができましたが、
 途中、いろいろと問題があり、ショートも起こり、たりして
 大変でした。しかし、無事完動させることができました。良かった
 です。イライラ棒も早く完動させたいです。
 これから新しい中1が入ってきますが、中1とはまず何でも
 いいから話すことが大切だと思います。入ってきたばかりの
 中1は何をすれば良いのかもわからないと思うので、こちら
 から話しかけてあげようと思います。
 文化祭では、中1にいろいろと教えてあげられるようにしま
 す。文化祭も頑張ります。



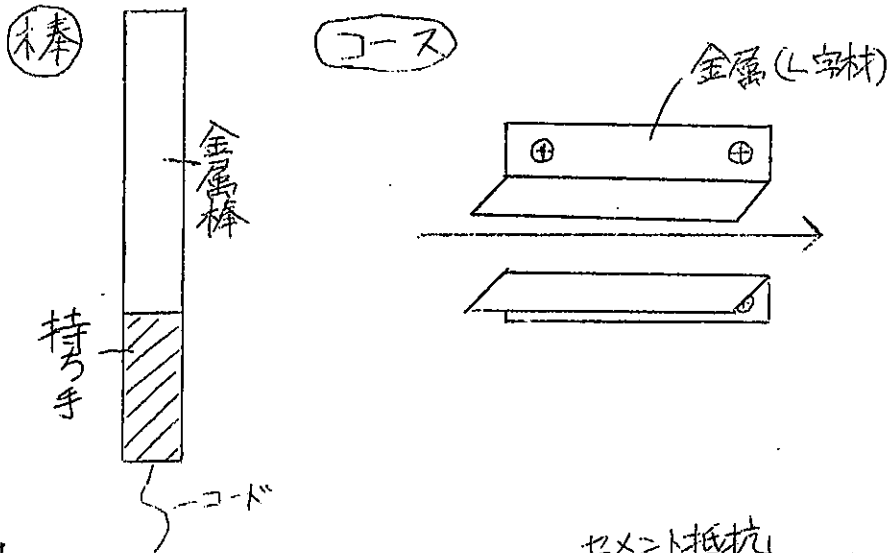
2012

イライラ棒です。3DSもはやる中、
2Dでのイライラ棒です。難しいので
頑張ってください。

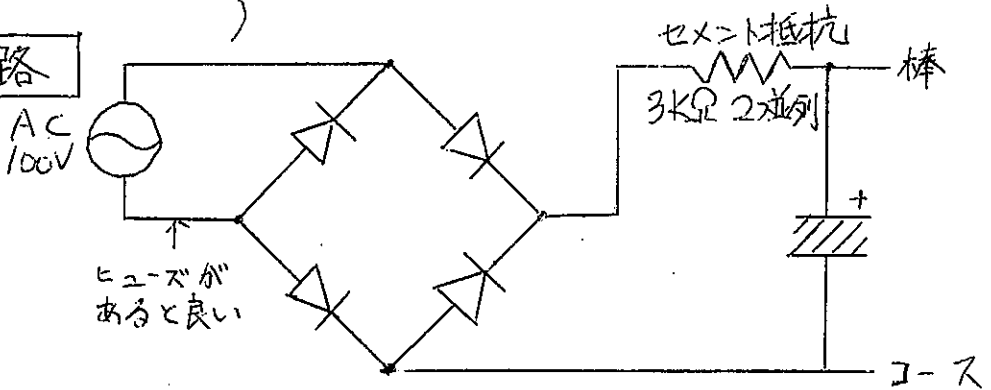
回路担当
M2 渡辺悠太
本体担当
M2 佐伯大地
協力
物無の皆様

ルール

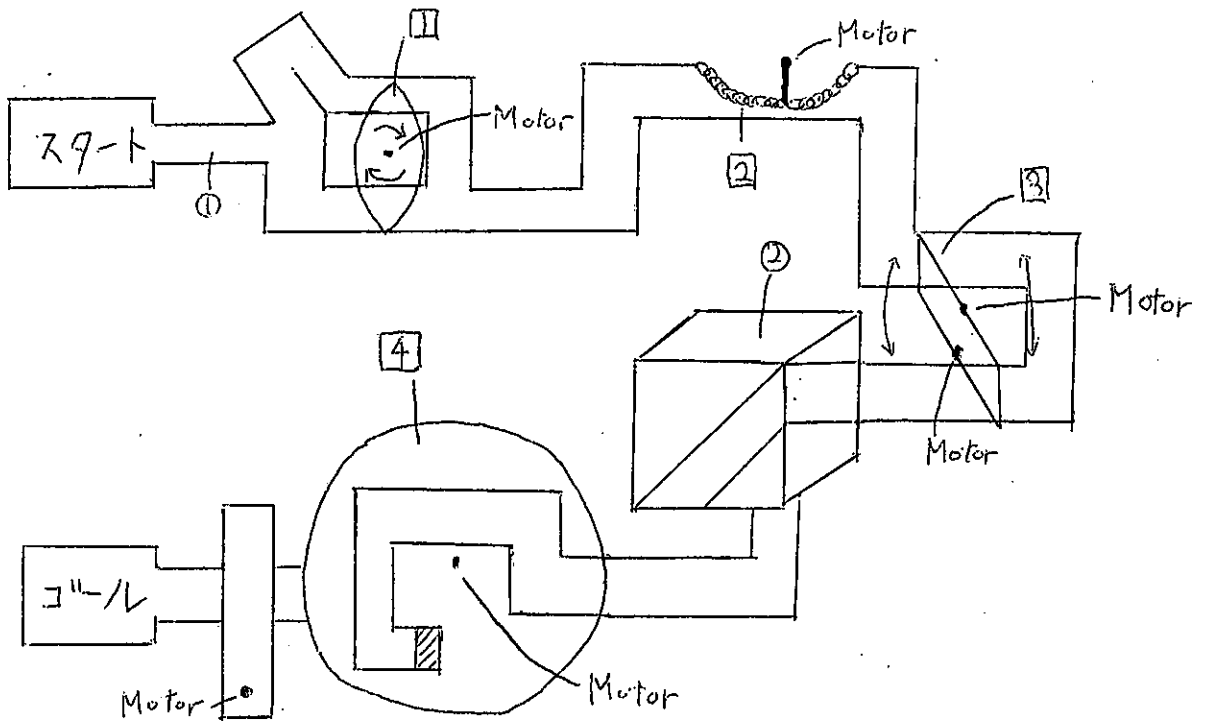
金属のできたコースがあるので、コースとコースの間に金属の棒を通していきましょう。棒とコースが触れると「フラッシュ」します。つまり、一気に電気が流れて「バチッ」と火花が散るわけです。左上がスタートで右下がゴールです。途中の障害物をくぐりぬけ、ゴールへ行きましょう。



回路



コース説明



コースポイント

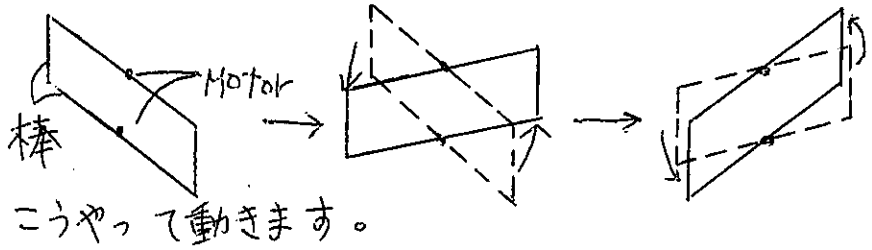
- ① スタート直後
細くなっています。気を付けましょう。
- ② 立体
上手に手首をひねって…。体ごと動かすのもアリ。
おちついて行きましょう。

ギミックポイント

- ① プロペラ状の金属
まわります。下よりも上のほうが乗かも…。
- ② フェーン
変わった動きです。でもおちついていれば大丈夫。

③ 平行四辺形

棒が上下すると思ってくればいいです。本当は



④ 田

最難関です。

入る → 中を通る → スイッチを押す → 中を通る → 外部に出る
をします。入るだけで大変です。中はおちつきながら
も素早く...

感想

佐伯

もう何本も糸ノコの歯を折ってしまいました。

すでに糸ノコ恐怖症です。

本体担当でありながらかなり渡辺に頼ってしま
い、本体担当らしいことはほとんどできません
でした。渡辺には感謝しています。

イライラ棒をつくるのは、木を切ったり、ねじ
を打ったり、ボンドでくっつけたりと大変では
したがとても楽しかったです。

イライラ棒をつくるにあたって、支えてくださ
った先輩方、物無の皆様、ありがとうございました。

渡辺

回路担当の渡辺です。回路担当になった訳は、
糸ノコが怖かったからです... 佐伯君と話が
かかっていますね。一日に三本も折れば流石に
恐怖症にでもなってしまう。

回路は動きません。難しいです。モーターは意
外と難しいです。今、2月下旬です。もうすぐ
3月にもなりますが、未だにファスターもつ
いていません。PICが壊れてしまって回路も今
は進んでいません。だんだん完動が怪しくな
ってきました。しかし、佐伯君に迷惑かけない
ように一生懸命頑張りたいと思います。
最後に一つ。

すげーイライラ棒にしてやる!!

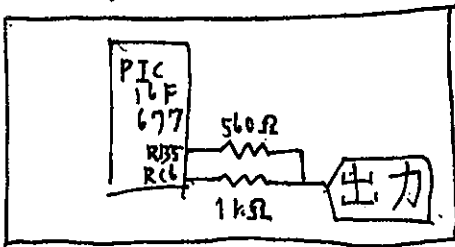
DEATH

STAR 2

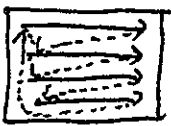
M2 石倉 匠
M2 横尾 幸丸
協力者 物無の 皆様

I ブラウソ管 (得点板用)

○ 回路図



○ しくみ



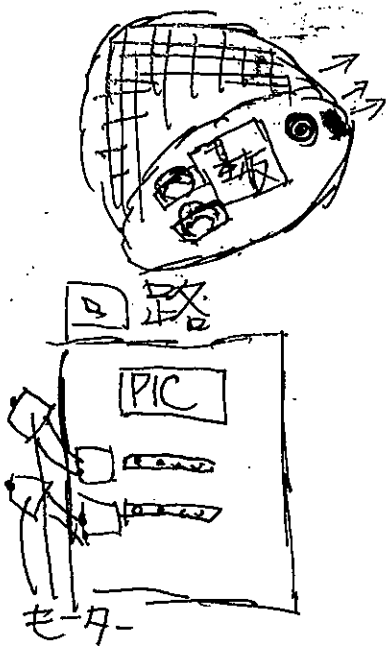
画面

光が図のように流れしており、それが1秒に60回というはやさで流れているので、画像が重なっているように見えるのです。

II 本体 (デススター)

モーター2つとタイヤとボールキャスターが地面につき、走り出します。
プログラムはC言語で行い、

図(見た目)



前進、後退、また、方向を
逆方向に回し、右方、
左方に移動できます。

また前の方に、色が変わると
止まり、後退するしくみを
つけました。

外装はなぜか「ザル」です。

感想

横尾

僕はあまり言語の事は教えてもらえずに、
プログラムなどをやるため、あまり進みませ
んでした。また、文書などのために、出席
回数も少なくなりました。しかし、新しいことを
学んで面白かったです。また頑張ります。

石倉

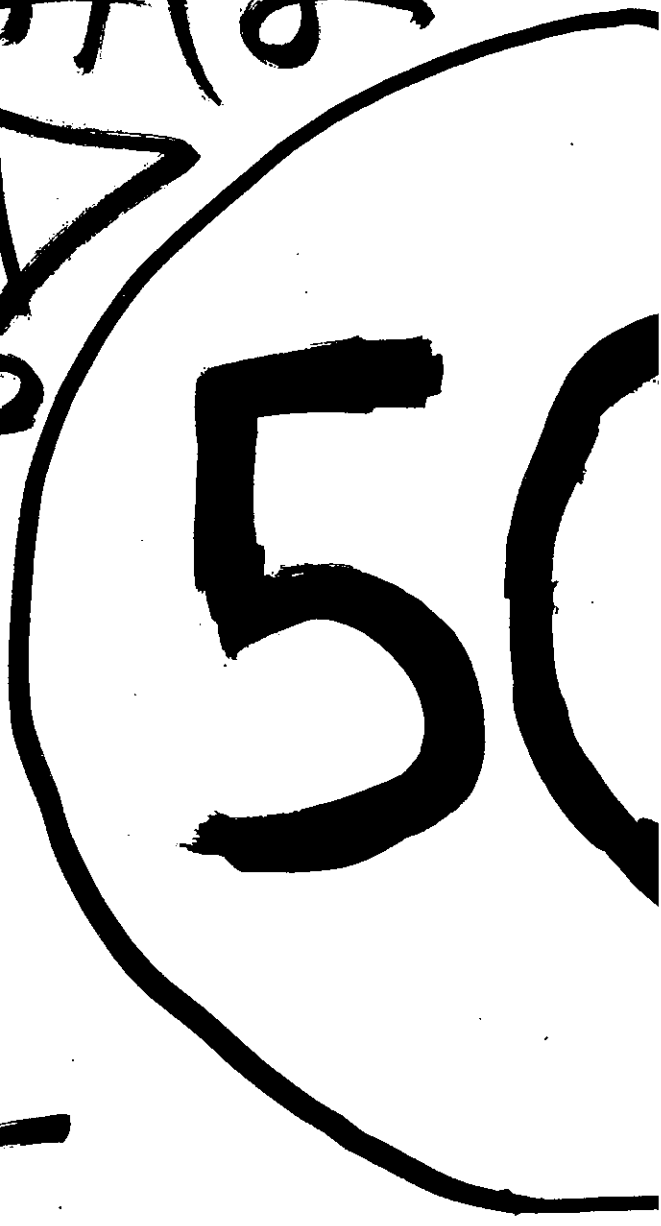
僕は、二回目の文化祭までにプログラミングを
やらせていたとき、しかもアセンブラもできたので
ラッキーでした。いろいろ事故もあり大変でしたが、
なにはともあれ楽しかったです。来年もプログラミングを
やりたいです。

500円は

最大10%

〜

まで



BUTURIS

制作:M3 佐藤 博

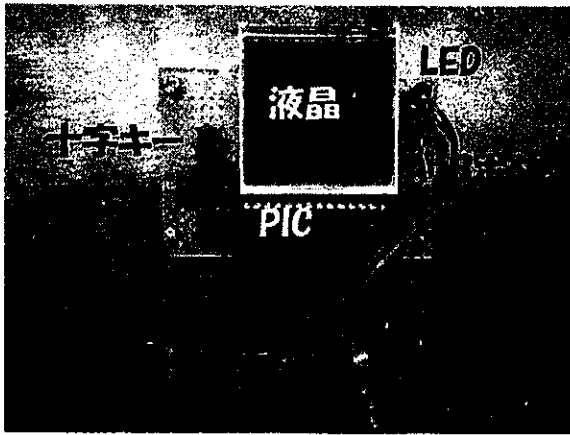
協力:物理部無線班のみなさん

✓ がいよう

物理部無線班では毎年売り物を販売させていただいているわけなのですが、今年は液晶に挑戦してみようということになりました、そこで動かすゲームとして知名度高き TETROS を選ぶことになりました。

※以下の文章は 2012/02/23 現在の制作状況に基づく記述でありますので、製品版の仕様と大きく異なる可能性があります。実際の製品の状態は購入される前に展示される(であろう)実機にてご体験されるか、物無員にお尋ねください。

✓ がいかん



●液晶

ゲーム画面が表示される液晶です。

今回は NOKIA3300 という携帯電話に使われている液晶を利用しています。

4096 色のきれいなカラー画像を表示してくれます。

制御チップが内蔵されており、PIC から色情報を送るだけで画面を表示してくれるというすぐれものです。

少しばかり反応速度が遅いのがネックで、画面が波打(遅延が起きる)ってしまいます。

画素 130×130pixel(なぜかコントローラーICは132×132、通販サイトは128×128pixelと認識していますが実際は130。)、正方形です。画素の密度としてはDS(3DSではなく)と同じくらい。

●PIC

このゲームの核となる部分です。CPUの役目もゲームカセットの役目もPICが担っています。

PIC16F1938というICを使っています。

周波数(一秒に計算できる回数は)32Mhzまでですが、実際に動かしているのは16Mhz。

ファミコンが1.79Mhz動作ですので、その約9倍、スーパーファミも3.58Mhzですので4倍近くの速度。このサイズにして優秀な計算装置です。ただ3DSが266Mhz×2、つまり532Mhz(実際にはもう少し下がる)ですので、最近のゲーム機の性能の高さがうかがえます。

●LED

試験型番にテバッグ(不具合消し)用に作りました。起動時にチカチカしたりしていました。

※実際の製品には搭載しない予定です。

●A/B ボタン、十字キー

ゲームを操作するためのボタンです。OETRIS以外のゲームもできるようにと十字+ABというボタン配置を選びました。ふぁ○こんのpak(py

●電池スナッフ

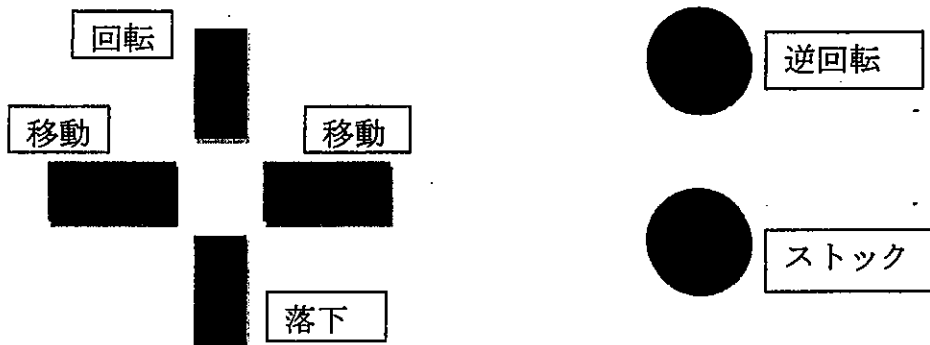
どこにでもある電池スナッフです。市販の9V電池を使用します。

裏側にある部品(レギュレータ)と共用することで3.3Vと9Vの二つの電圧を取り出しています。

✓ そうさほうほう

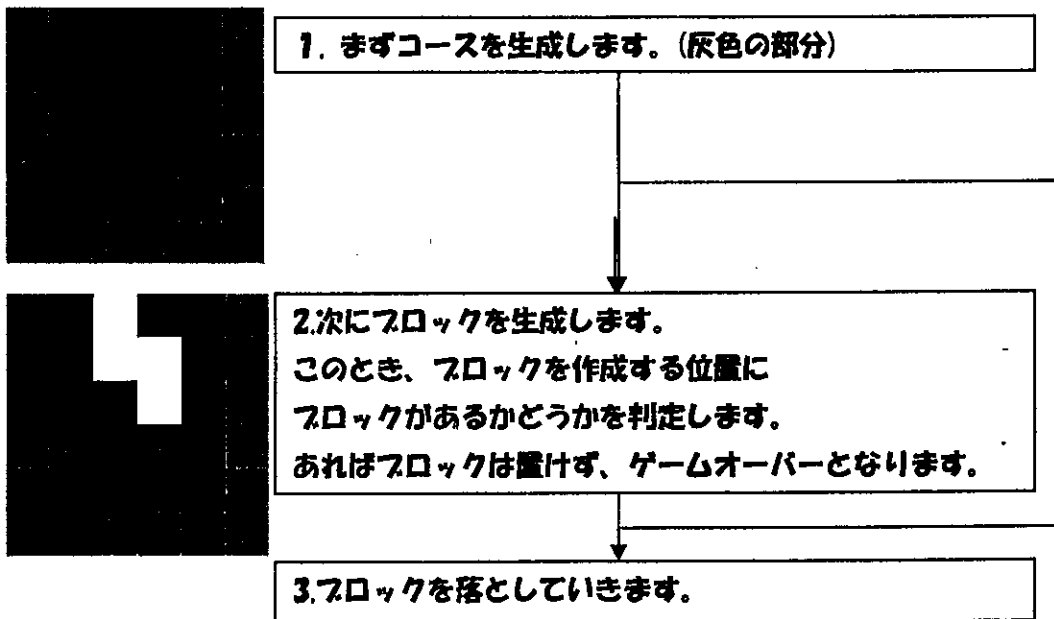
本家TOTRISとそろえる予定です。ストック(ホールド)は実装しないかもしれません。

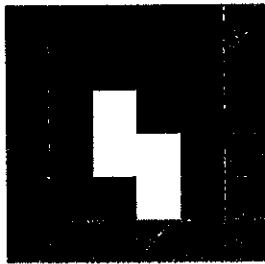
詳しくは製品版のマニュアルをご覧ください。



✓ ぷろぐらむ

HITEC-C コンパイラを使い、C 言語でプログラミングをしています。何の参考にもなりませんがプログラム約 1000 行程の長さです(効率が悪い)。液晶の制御部分はインターネットを参考にさせていただき、ここでは TEORIS 自体のプログラムを解説させていただきます。

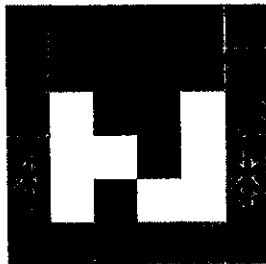




4.一段落ちるたびにブロックの下にブロックがあるか判定します。
ここでブロックがあれば下に衝突したものととしてブロックは停止します。



5.ブロックが衝突すると揃った列がないか調べます。
ここでは左のように揃ったとします。



6.揃っていた場合はその列を消去し、そこより上の列を一段下に落とします。
2列以上揃った場合を考え、5に戻り繰り返します。

7.点数を計算します。
一列揃えば+1。もし4列が1回で揃えばコンボも入ります。

✓ かんそう

本家の制作(T6)でサーボ制御がうまくいかなかった分、プログラムを書き換えるだけでゲームが動作するこちらの制作は気楽に楽しむことができました。

PICの容量と格闘する毎日ですが、どうにかして発売までにはほかのゲームも詰め込み、多機能ゲーム機として売り出せるようにしたいと思っています。

制作の手助けをしてくださった物理部無線班の皆様にこの場をお借りしてお礼を申し上げます。

チカチカ

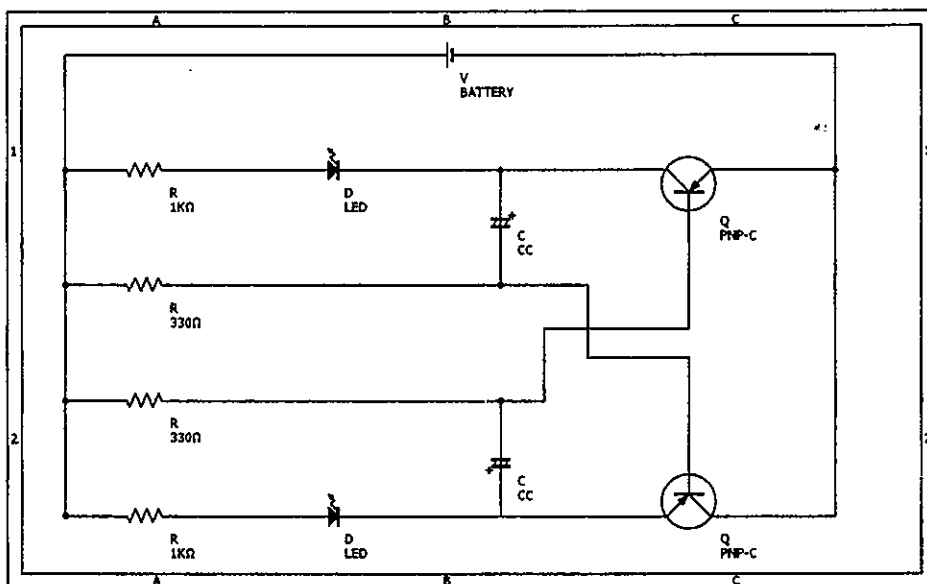
M3 ぐんぼ 野口 英世

・チカチカとは

チカチカ(本名=自走マルチバイブレータ)とは、二つのLEDという物が交互に輝く、その名の通りの物です。その原理は、電池から流れた電気がコンデンサーという物に溜まり、その電気がLEDに流れ、それが二つ交互に行われる、というものです。容量の違うコンデンサーを使えば、点滅の速度も変えられる訳です。

仕組みは何だか難しそうに見えますが、製作は半田ごてを持った事の無い方でも可能です。

君とは一生で会えない、僕がいる時には、君はもういない——



↑回路図です。

麻布学園物理部無線班

回路図集 2012

発行日	:	2012/04/26
編集責任者	:	新田 京太郎
印刷責任者	:	栗本 郁也
デザイン	:	岸田 聖生
	:	水野 太郎
	:	山本 涼一
	:	花園 佳月

乱丁・落丁はお取替え致します。

転載・複製は自由です。

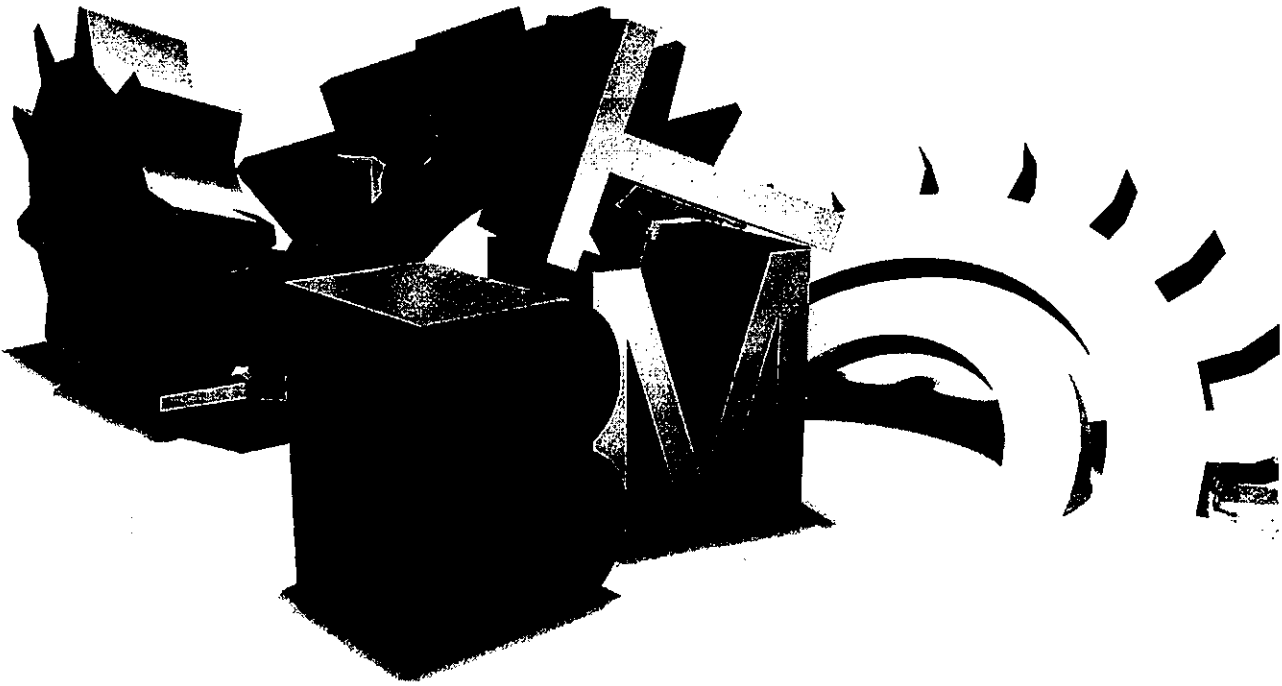


物理部無線班 Web



物理部無線班 Youtube

TAKE FREE



2012 AZABU FESTIVAL
PHYSICS RADIO CLUB